Ääniliitännäisten ohjelmointi, FL Studio ja sen vertailu muihin ääniohjelmistoihin

HAMK
HÄMEEN AMMATTIKORKEAKOULU
HÄME UNIVERSITY OF APPLIED SCIENCES

Ammattikorkeakoulututkinnon opinnäytetyö

Hämeen ammattikorkeakoulu, Tietojenkäsittelyn koulutusohjelma

Visamäki, syksy 2017

Valtteri Tuominen

HÄMEEN AMMATTIKORKEAKOULU
HÄME UNIVERSITY OF APPLIED SCIENCES

VISAMÄKI
Tietojenkäsittelyn koulutusohjelma

Tekijä Valtteri Tuominen **Vuosi** 2017

Työn nimi Ääniliitännäisten ohjelmointi, FL Studio ja sen vertailu muihin ääniohjelmistoihin

Työn ohjaaja Lasse Seppänen

TIIVISTELMÄ

Opinnäytetyön tavoitteena on rakentaa toimiva äänenvoimakkuutta säätelevä liitännäinen, perehtyä erittäin syvällisesti FL Studio -ääniohjelmistoon sekä tutkia sen mahdollisia eroavuuksia ja sekä työn jälkeä muihin sitä vastaaviin ääniohjelmistoihin verrattuna.

Opinnäytetyön viimeisessä osassa käsitellään vertailu muista ääniohjelmistoista, jotka ovat Adobe Audition, Ableton Live 9 Lite sekä Pro Tools First. Työssä pyritään havainnollistamaan mahdollisimman tehokkaasti ohjelmistojen eroavuuksia.

Tutkimustulokset antoivat ilmi, että eri ohjelmistojen välillä saaduissa arvoissa oli eroja, mutta vain sellaisia joita ihmiskorva ei ollenkaan havaitse. Tutkimustulokset havainnollistettiin taulukkoina, jotka antavat osviittaa ohjelmistojen eroavuudesta. Suurimmat eroavuudet tulivat ilmi niin ohjelmistojen välillä, kuin sekä tiettyjen taajuusasteikoiden kohdalla.

Avainsanat Ableton Live, Adobe Audition, FL Studio, Pro Tools, ääniohjelmisto, vertailu, JUCE, ääniohjelmointi, ääniliitännäinen

Sivut 55 sivua, joista liitteitä 8 sivua

HÄMEEN AMMATTIKORKEAKOULU
HÄME UNIVERSITY OF APPLIED SCIENCES

VISAMÄKI
Degree Programme in Business Information Technology

Author Valtteri Tuominen **Year** 2017

Subject Audio Plugin Programming, FL Studio And Comparison with Other DAW-Programs

Supervisor Lasse Seppänen

ABSTRACT

The goal of the thesis is to build a functional audio gain program, explore the FL Studio -program in detail and investigate its possible differences in comparison with other DAW-programs.

In the last part of the thesis, the FL Studio is compared with the other DAW-programs, which are Adobe Audition, Ableton Live 9 Lite and Pro Tools First. The aim of the comparison is to efficiently demonstrate the differences between these programs.

The research results indicate that there are differences between the programs, but only the kind that the human ear is not capable of hearing. The results were demonstrated with charts, which reveal the differences between the programs. The biggest differences were detected when comparing the programs with one another, and within certain frequency ranges.

Keywords Ableton Live, Adobe Audition, FL Studio, Pro Tools, DAW-program, comparison, JUCE, audio programming, audio plugin

Pages 55 pages including appendices 8 pages

SISÄLLYS

1 JOHDANTO

Ääniliitännäisten ohjelmointi on aiheena hieman tuntematon. Etsiessään tietoa ja ohjeistuksia internetistä käyttäjä saa varautua erittäin vähäiseen informaatioon. Äänitekniikka kehittyy, joten uusia tapoja ääniprojektin rakentamiseen ja sen helpottamiseen on odotettavissa tulevaisuuden myötä. Tämä mahdollistaa myös uusien kehittäjien astumisen esiin ääniliitännäisten ohjelmoinnin maailmassa.

FL Studio -ääniohjelmisto on työn suurin keskittymiskohde eri ohjelmistoihin verrattaessa. Kokemusta tämän tyyppisistä ohjelmistoista on kertynyt muutaman vuoden ajalta. Aloittelevan käyttäjän suositellaan opiskelevan erilaista materiaalia mediasta ja erilaisista palveluista internetistä. Ääniprojektin rakentamiseen ei ole tiettyä järjestystä tai kaavaa, mutta toki kokeneempien käyttäjien ehdotukset ja ohjeet kannattaa ottaa huomioon. Yksi tärkeimmistä asioista ääniprojektin rakentamisen opiskeluun ja käyttäjän itsensä kehittämiseen löytyy korvien välistä eli käyttäjän omasta päästä. Projektien aikana suositellaan pitämään taukoja, sillä korvat alkavat valehdella liian useiden työtuntien aikana. Käyttäjän kannattaa myös ottaa talteen niin huonompia kuin parempiakin projekteja. Ne auttavat käyttäjää eteenpäin ja antavat lisää kokemusta seuraaviin ja sitä seuraaviin projekteihin.

Työssä pyritään havainnollistamaan mahdollisimman tehokkaasti ohjelmistojen eroavuuksia niihin kohdistuvan vertailun kautta. Lähtökohtaisesti mikä tahansa seuraavista ohjelmistoista voi olla käyttäjän mielestä se oikea, kyse on erittäin pitkälti käyttäjän mielipiteistä. Työssä tutkitaan myös eri ohjelmistojen antamia lopputuloksia. Onko ohjelmistojen tuottamassa lopputuloksessa eroavuuksia? Jos on, ovatko ne suuria vai pieniä?

2 ÄÄNILIITÄNNÄISTEN OHJELMOINTI

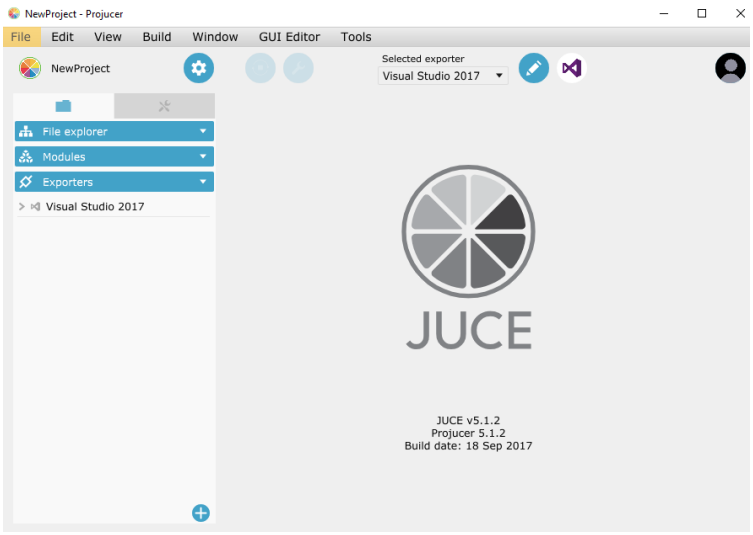
Työn alkuvaiheessa perehdytään ääniohjelmistoissa käytettävien liitännäisten ohjelmointiin. Tavoitteena on rakentaa toimiva ohjelma, jolla hallitaan rakennetun äänitiedoston äänenvoimakkuutta FL Studio - tai sitä vastaavien ohjelmistojen sisällä. Ohjelman varsinainen koodaus tapahtuu Visual Studio 2017 -ohjelmankehitysympäristössä käyttäen C++-koodikieltä, mutta apuohjelmana käytetään JUCE-ohjelmistoa.

2.1 JUCE-ohjelmisto

JUCE (Jules' Utility Class Extensions) on avoimen lähdekoodin ohjelmisto, jonka on kehittänyt yhtiö nimeltä ROLI. JUCE on luotu ääniliitännäisten rakentamisen helpottamiseksi sisältäen kirjaston, johon on asetettu valmiita luokkia ja tiedostoja ääniliitännäisen rakentamista varten. Projekti alustetaan JUCE-ohjelmistossa, joka ohjaa sen Visual Studio-ohjelmankehitysympäristöön. JUCE rakentaa ohjelmalle ulkoisen sekä sisäisen alustuksen, jonka pohjalta käyttäjällä on paljon yksinkertaisempi lähtökohta ohjelmansa rakentamiseen. (JUCE n.d-a.)

JUCE-ohjelmiston verkkosivustoilta on mahdollista ladata ohjelmisto eri versioineen sekä tarkastella valmiiksi luotuja luokkia sekä niiden tietoja. Ohjelmisto sisältää ilmaisen tai maksulliset versionsa. Ilmaisessa versiossa JUCE näyttää hetkellisesti logonsa valmiin ääniliitännäisen käynnistyessä, jonka saa poistettua tilaamalla maksullisen version. (JUCE n.d-d.)

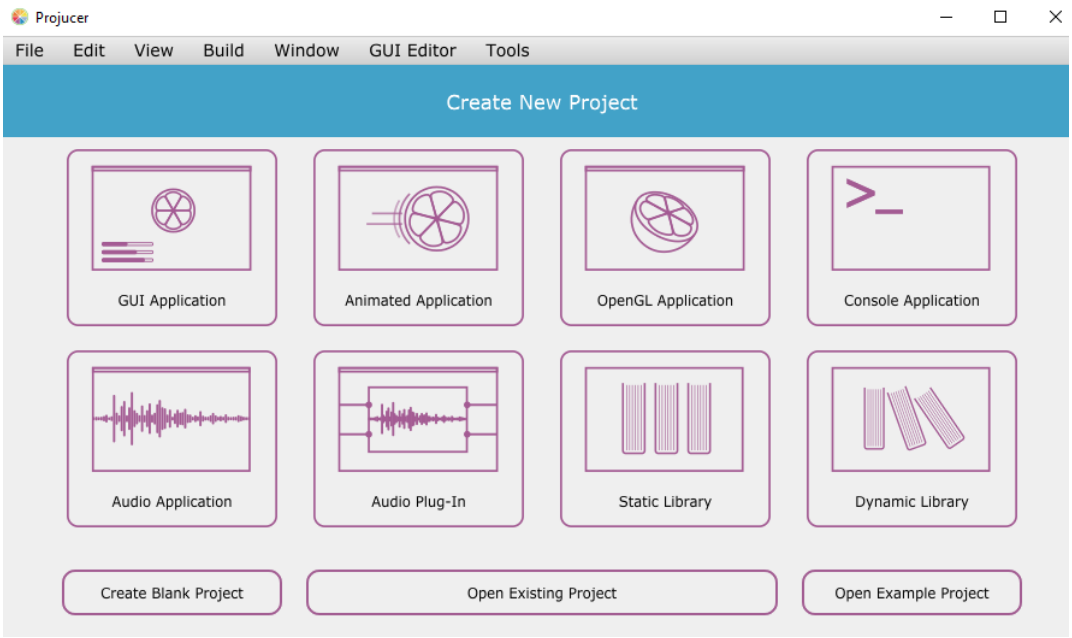
JUCE-ohjelmistoa ovat käyttäneet ääniliitännäistensä luomiseen myös tunnetut valmistajat, joita ovat mm. Image Line, joka on kehittänyt tässä työssä suuressa osassa olevan FL Studio -ohjelmiston. Alla sijaitsevassa kuvassa (ks. Kuva 1) nähdään JUCE-ohjelmiston esimerkkiprojekti. (JUCE n.d-b.)



Kuva 1. JUCE-ohjelmiston esimerkkiprojekti.

2.2 Projektin alustus

Projekti alustetaan luomalla uusi projekti JUCE-ohjelmistossa. Ohjelmistossa on valittavissa useita erilaisia alustuksia projekteille tai käyttäjä voi aloittaa projektinsa rakentamisen myös täysin tyhjältä pohjalta, kuten alla sijaitsevassa kuvassa näkyy (ks. Kuva 2). Työssä lähdetään rakentamaan VST-pohjaista ohjelmaa, joten tässä tapauksessa valitaan ”Audio Plugin” -tyyppinen projekti.

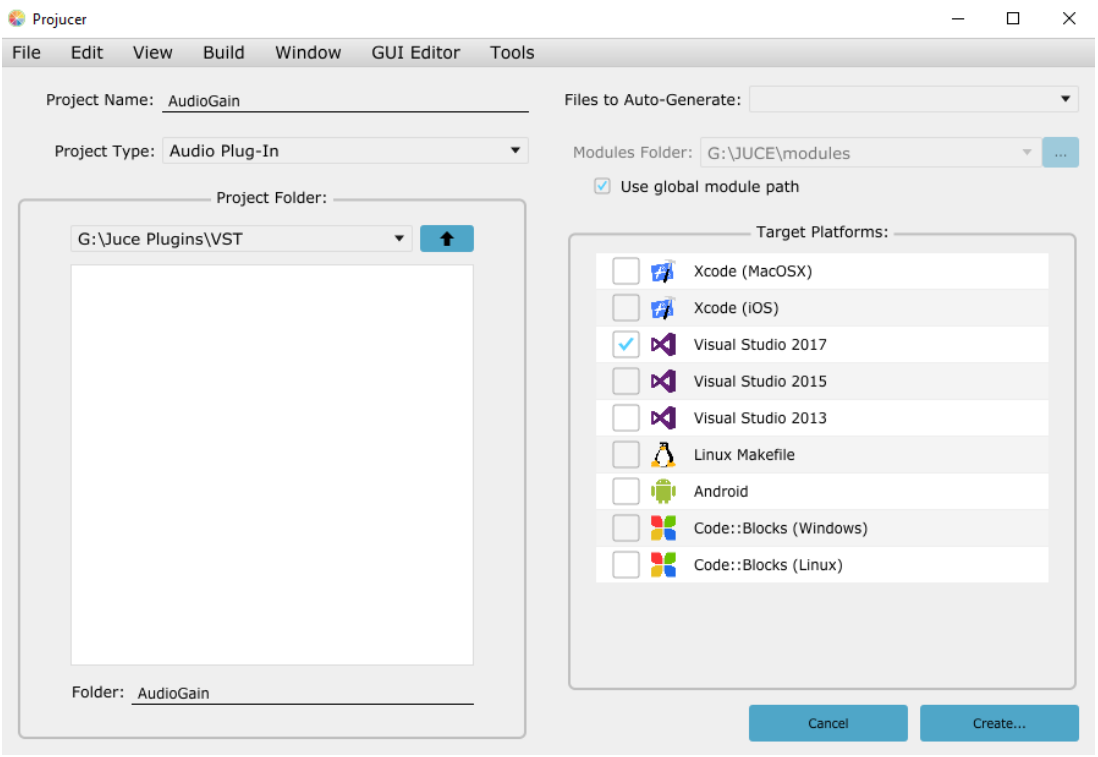


Kuva 2. JUCE-ohjelmistossa valittavat esimerkkiprojektit.

2.2.1 Projektin kohdekansion sekä ohjelmankehitysympäristön määrittäminen

Seuraavassa vaiheessa määritetään projektin kohdekansio sekä ohjelmankehitysympäristö, jolla käyttäjä aloittaa projektinsa rakentamisen. Alla sijaitsevassa kuvassa on määritelty ohjelman kohdekansio, ohjelman nimi (”AudioGain”) sekä käyttämään Visual Studio 2017 -ohjelmankehitysympäristöä (ks. Kuva 3).

Kuten alla sijaitsevasta kuvasta havaitaan, käyttäjän on mahdollista käyttää JUCE-ohjelmistoa monessa eri ohjelmankehitysympäristössä niin Windows-, Linux- kuin iOS-järjestelmissä. Valittavissa on myös Android-pohjainen ohjelma, joten käyttäjällä on mahdollista JUCE-ohjelmistossa rakentaa ohjelma myös Android-järjestelmiin Android Studio -ohjelmankehitysympäristön avulla.



Kuva 3. Kohdekansion ja ohjelmankehitysympäristön määrittäminen uuden projektin alustuksessa.

2.2.2 Projektin aloitusnäkymä JUCE-ohjelmistossa

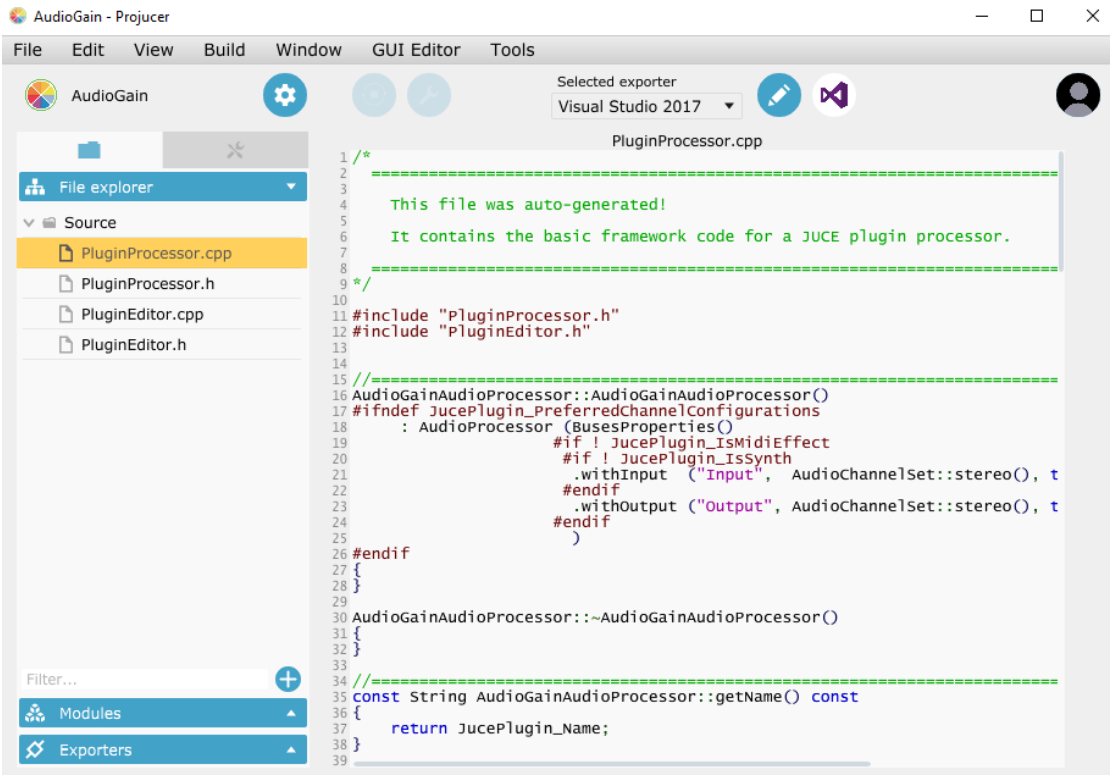
Projektin luomisen jälkeen JUCE-ohjelmisto rakentaa automaattisesti sille tarvittavat kansiot ja tiedostot käyttäjän määrittämään hakemistoon. Alla sijaitsevassa kuvassa nähdään projektin aloitusnäkymä. JUCE-ohjelmisto on alustanut muutaman erilaisen tiedoston, joita ohjelman rakentamisessa tarvitaan. Ne ovat alla sijaitsevassa kuvassa nähtävät ”PluginProcessor.cpp”, ”PluginProcessor.h”, ”PluginEditor.cpp”, ”PluginEditor.h” (ks. Kuva 4).

Projektin luonne määrittää näiden tiedostojen määrän ja rakenteen. VST-pohjaista ohjelmaa rakentaessa JUCE-ohjelmisto määrittää nämä kyseiset tiedostot. Jokaisen tiedoston koodia on mahdollista muuttaa myös JUCE-ohjelmistossa, mutta ohjelman testauksen kannalta suositellaan muutoksia tehtävän ohjelmankehitysympäristön sisällä.

Projektissa on mahdollista muokata erilaisia asetuksia, joista ei ole tarvetta muuttaa kuin ainoastaan yhtä (ks. Kuva 5), jos käyttäjä rakentaa VST3-pohjaista ääniliitännäistä. Tämä on kriittinen muutos ohjelman toimivuuden kannalta. Kun taas käyttäjän rakentaessa VST-pohjaista liitännäistä, tämä muutos ei vaikuta ohjelman toimivuuteen.

Seuraavaksi käyttäjä avaa JUCE-ohjelmiston alustaman ohjelman määrittämässään ohjelmankehitysympäristössä, joka on tässä tapauksessa Visual

Studio 2017. Ohjelmankehitysympäristö avautuu JUCE-ohjelmiston yläkulmassa näkyvästä Visual Studio -logosta. Ohjelmankehitysympäristön avautuessa se rakentaa projektille sen tarvitsemat tiedostot.



Kuva 4. Projektin aloitusnäkymä projektin luomisen jälkeen.

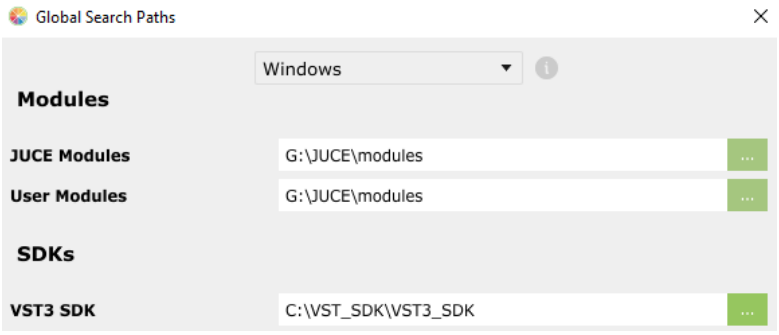
2.2.3 VST SDK3 -ohjelmankehityspaketin määrittäminen

JUCE-ohjelmisto vaatii toimiakseen Steinberg-yhtiön kehittämän VST SDK3 -ohjelmankehityspaketin (*engl. Software Development Kit*) lataamisen työasemalle, joka on luotu liitännäisten kehittäjiä varten. Ohjelmankehityspaketti on ladattavissa Steinberg-yhtiön verkkosivustoilta. (Steinberg n.d-a.)

Käyttäjä sijoittaa paketin työasemalleen ja määrittää sen JUCE-ohjelmistossa kohdasta ”File – Global Search Paths”, kuten alla sijaitsevassa kuvassa on määritelty. Ilman kyseistä ohjelmankehityspakettia Visual Studio -ohjelmankehitysympäristö antaa virheilmoituksia, mikä taas aiheuttaa ongelmia VST3-pohjaisen ääniliitännäisen rakentamisessa sekä testauksessa. Kyseisessä ohjelmassa käytetään ainoastaan VST-pohjaista ääniliitännäistä, joten ongelmia ei tässä tapauksessa synny, vaikka ohjelmankehityspakettia ei olisi määriteltykään.

VST3-pohjainen ääniliitännäinen on uudempi sekä kehittyneempi versiomuoto vanhemmasta VST-pohjaisesta ääniliitännäisestä. VST3-pohjainen liitännäinen on hieman kevyempi suorittaa, sekä ne eroavat myös tiedostomuodoissaan. VST-liitännäinen toimii ”.dll”- muotoisella tiedostotyypillä

(esim. AudioGain.dll), kun taas VST3-liitännäisen tiedostotyyppi on ”.vst3” (esim. AudioGain.vst3).



Kuva 5. VST3 SDK- ohjelmankehityspaketin määrittäminen JUCE-ohjelmiston asetuksissa.

2.3 Äänenvoimakkuutta säätelevän ohjelman rakentaminen Visual Studio 2017 -ohjelmankehitysympäristössä

JUCE-ohjelmisto luo automaattisesti käyttäjän ohjelmalle sen tarvitsemat tiedostot. Kuten alla sijaitsevassa kuvassa näkyy (ks. Kuva 6), se sisältää lukuisia JUCE-ohjelmiston kehittäjien rakentamia tiedostoja. Jokaiseen tiedostoon on lisätty kehittäjien alustamia kommenttirivejä, joissa on selvennetty koodin toimintaa.

Seuraavan vaiheen tarkoituksena on rakentaa ohjelma, joka säätelee äänenvoimakkuutta FL Studio - tai muissa vastaavissa DAW-ääniohjelmistoissa. Työn kokonainen koodi on sisällytetty liitteisiin (ks. Liite 1, Liite 2, Liite 3 ja Liite 4) sekä seuraavissa aihealueissa käydään lävitse muutamia pätkiä itse lisätyistä koodeista. Käyttäjä voi halutessaan perehtyä JUCE-ohjelmiston rakentamiin koodeihin itsenäisesti, jos kokee sen tarpeelliseksi.

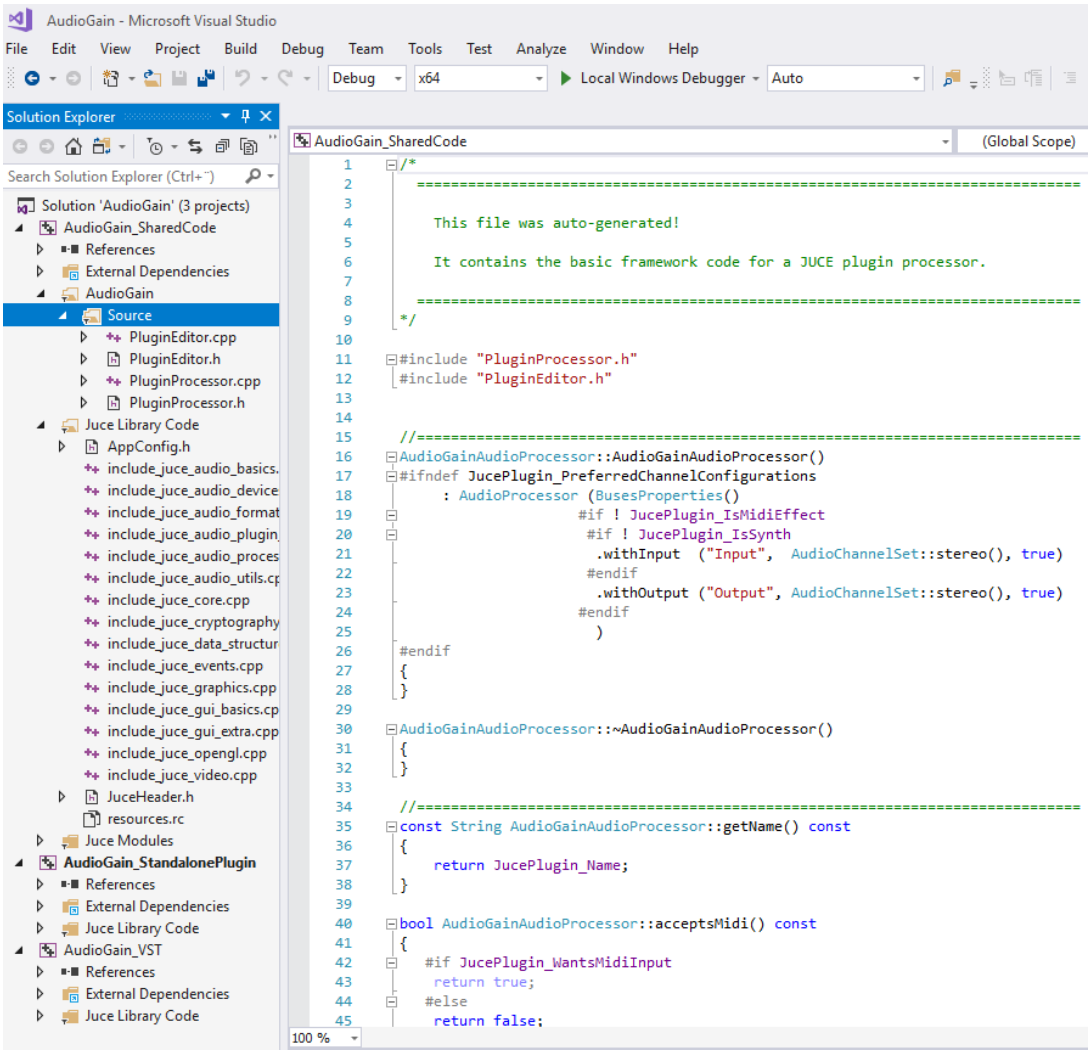
Seuraavat osiot tulevat sisältämään lyhyitä selvennyksiä tiedostoihin (”PluginProcessor.cpp”, ”PluginProcessor.h”, ”PluginEditor.cpp” ja ”PluginEditor.h”) lisätyistä koodeista ja äänenvoimakkuuden säätelyyn vaikuttavat koodit ovat kommentoituna (”**//**”) sekä **lihavoituina** liitteissä. Eri aihealueiden määrityksiin on lisätty ohjelman tulkintaa helpottavia selvennyksiä. Ohjelman rakentamiseen on käytetty apuna videoita sekä muita verkosta löytyviä tutoriaaleja. (JUCE n.d-c.)

Äänenvoimakkuutta säätelevän ohjelman rakentamiseen liittyy muutamia aihealueita, joita ovat äänenvoimakkuuden säätimen graafinen käyttöliittymä, äänen prosessointi sekä säätimen ja äänen välillä tapahtuvien virheiden korjaaminen, automaatiotiedon asettaminen ohjelman käyttöön sekä edellisen arvon tallentaminen ohjelman muistiin.

2.3.1 Projektin aloitusnäkymä Visual Studio 2017 -ohjelmankehitysympäristössä

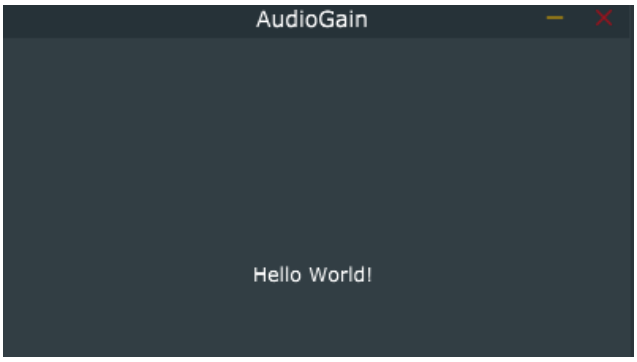
JUCE-ohjelmisto rakentaa kolme erilaista projektia (ks. Kuva 6). ”AudioGain_Shared_Code”-projektissa tapahtuu itse ohjelman muokkaus sekä sen rakentaminen. ”AudioGain_StandalonePlugin”-projekti määrittää ohjelmalle ”.exe”- pohjaisen tiedoston, joka on käytettävissä Windows-käyttöjärjestelmässä erillisenä ohjelmanaan. ”AudioGain_VST”-projekti taas määrittää ”.dll”- pohjaisen tiedoston, jota käytetään liitännäisenä FL Studio - tai muissa vastaavanlaisissa DAW-ohjelmistoissa.

Visual Studio luo tiedostot kansioon, johon käyttäjä on ne aikaisemmassa vaiheessa määrittänyt. Ohjelman rakentaminen suoritetaan ”PluginEditor”- ja ”PluginProcessor”-tiedostojen sisällä. ”PluginEditor”-tiedostot määrittävät ohjelmalle graafisen, eli ulkoisen puolen, kun taas ”PluginProcessor”-tiedostot määrittävät ohjelman sisäisen rakenteen, eli ohjelman toimivuuden. Muihin JUCE-ohjelmiston luomiin tiedostoihin ei ole tarvetta koskea tämän ohjelman rakentamisen yhteydessä.



Kuva 6. JUCE-ohjelmiston automaattisesti luoma ohjelma Visual Studio 2017 -ohjelmankehitysympäristössä.

Käyttäjän suorittaessa ohjelman ensimmäistä kertaa ilman minkäänlaisia muutoksia koodiin, se näyttää ainoastaan tekstin ”Hello World”, kuten alla sijaitsevassa kuvassa on havainnollistettu (ks. Kuva 7).

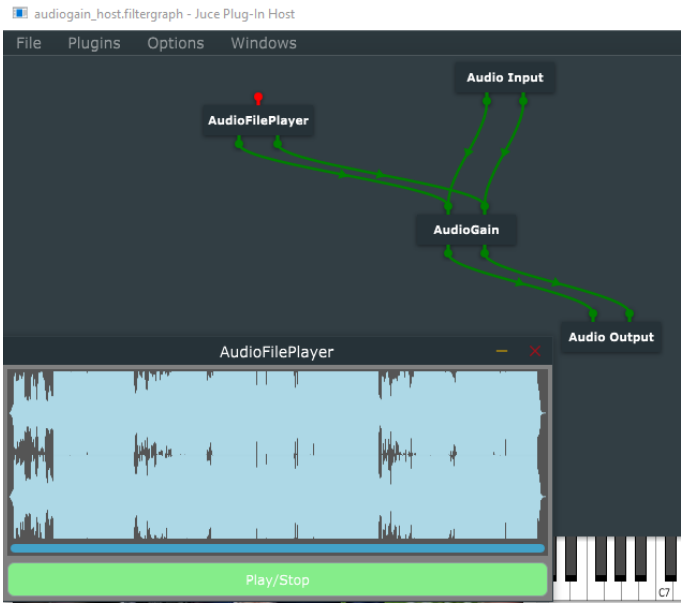


Kuva 7. Ohjelman ajaminen ensimmäistä kertaa.

2.3.2 Ohjelman testaus JUCE-ohjelmistoon sisältyvällä työkalulla

JUCE-ohjelmistoon sisältyy työkalu, jonka tarkoituksena on helpottaa ohjelman testausta ohjelmankehitysympäristön sisällä (ks. Kuva 8). Käyttäjän ei siis tarvitse käydä testaamassa ohjelmaansa FL Studio -ohjelmistossa jokaisen muutoksen jälkeen. Sen toiminta perustuu äänitiedoston toistamiseen työkalussa, joten käyttäjä voi havainnoida muutoksia tämän työkalun sisällä. Työkalu löytyy JUCE-ohjelmiston ”Examples”-kansiosta.

Plugin Host -työkalu määritetään ohjelmankehitysympäristöön testauksen asetuksista (Visual Studio 2017 -ohjelmankehitysympäristössä ”Debug – Plugin Properties – Debugging”). Käyttäjä asettaa Plugin Host -työkalun tiedostonpolun ”Command”-riville, joten työkalu käynnistyy käyttäjän ajaessa ohjelman sen kehitysympäristössä.



Kuva 8. Plugin Host-työkalu.

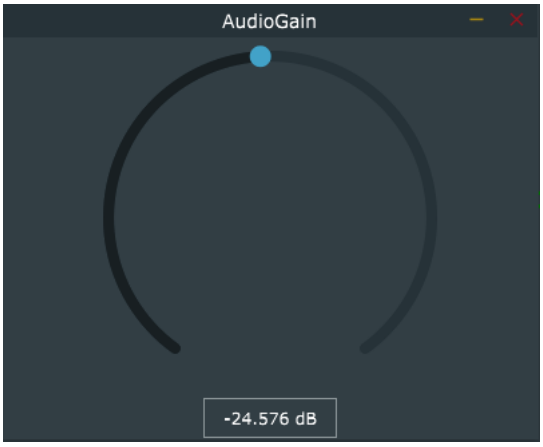
2.3.3 Äänenvoimakkuuden säätimen graafinen käyttöliittymä

Alustavasti on lähdettävä liikkeelle siitä, miltä ohjelma näyttää graafiselta osuudeltaan. Tavoitteena on rakentaa ohjelmaan yksinkertainen säädin, joka säätelee äänenvoimakkuutta ja näyttää erilaisia arvoja säätimen alapuolella sijaitsevassa teksti-ikkunassa. Alla sijaitsevat määritykset vaikuttavat graafiseen käyttöliittymään ja alla sijaitsevassa kuvassa näkyy lopputulos (ks. Kuva 9).

Koodien positiot eri tiedostoissa on nähtävissä liitteissä (ks. Liite 1 ja Liite 2), joista ohjelmaan lisättyjä kyseisiä koodeja on helpompi tulkita. Alla sijaitseva taulukko sisältää äänenvoimakkuuden säätimen graafisen käyttöliittymän määritykset (ks. Taulukko 1).

Taulukko 1. Graafisen käyttöliittymän määritykset.

gainSlider	Säätimen määrittäminen
setSize	Ikkunan koko
setSliderStyle	Säätimen tyyli
setBounds	Säätimen koko
setRange	Säätimen arvoalue
addAndMakeVisible	Säätimen näkyvyys ikkunassa
setTextBoxStyle	Teksti-ikkunan koko ja tyyli
setTextValueSuffix	Arvon perään teksti ”dB”



Kuva 9. Äänenvoimakkuuden säädin.

2.3.4 Äänen prosessointi sekä säätimen ja äänen välillä tapahtuvien häiriöiden korjaaminen

Seuraavaksi tehdään muutokset äänen prosessointiin. Desibeliasteikko muodostuu kaavan pohjalta, sillä desibeliasteikko on logaritminen. Koodiin

on sisällytetty myös korjauksia äänen ja säätimen välille, sillä ääni saattaa ottaa häiriöitä säätimen arvojen nopeissa muutoksissa.

Koodien positiot eri tiedostoissa on nähtävissä liitteissä (ks. Liite 1, Liite 2, Liite 3 ja Liite 4), joista ohjelmaan lisättyjä kyseisiä koodeja on helpompi tulkita. Alla sijaitseva taulukko sisältää äänen prosessoinnin sekä säätimen ja äänen välillä tapahtuvien häiriöiden korjaamisen määritykset (ks. Taulukko 2).

Taulukko 2. Äänen prosessoinnin sekä säätimen ja äänen välillä tapahtuvien häiriöiden korjaamisen määritykset.

gainValue	Muuttuja
previousGain	Muuttuja
parameters	Olio
Slider::Listener	Luokka säätimen kuuntelijalle
GAIN_ID	Makro parametrien määrittämiseen
GAIN_NAME	Makro parametrien määrittämiseen
addListener	Toimintoja säätimen arvojen muuttuessa
sliderValueChanged	Toimintoja säätimen arvojen muuttuessa
previousGain	Kaava desibeliasteikon määrittämiseen
currentGain	Kaava desibeliasteikon määrittämiseen
currentGain == previousGain	Äänen häiriöiden estäminen
GainRamp	Äänen häiriöiden estäminen

2.3.5 Automaatiotiedon asettaminen ohjelman käyttöön

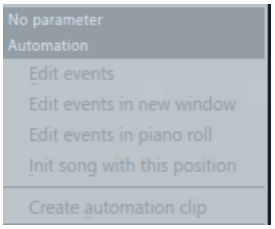
Äänenvoimakkuutta säätelevä ohjelma tarvitsee myös osion automaationtiedon käsittelyyn. Automaatio on siis tapa, jolla voidaan helpottaa ja automatisoida äänen kulkua FL Studio - tai sen kaltaisissa ääniohjelmistoissa. Ääniohjelmistossa on viiva, joka mittaa arvoja ääniliitännäiseen asetettujen minimi- ja maksimiarvojen välillä. Tätä viivaa säätelemällä ylös tai alas, saadaan automaatio säätelemään arvoja automaattisesti aikajanalla. Tässä tapauksessa, säätelemään äänenvoimakkuutta automaattisesti käyttäjän asettelemiin arvoihin.

Tämä vaatii oman toimintonsa ääniliitännäisen rakentamisessa, sillä ilman tätä toimintoa käyttäjä ei voi käyttää automaatiotietoa ääniohjelmistossa. FL Studio -ääniohjelmiston tapauksessa kyseisen automaatiotiedon asettaminen käyttöön ei ole edes mahdollista siihen tarkoitetusta painikkeesta (ks. Kuva 10).

Koodien positiot eri tiedostoissa on nähtävissä liitteissä (ks. Liite 1, Liite 2 ja Liite 3), joista ohjelmaan lisättyjä kyseisiä koodeja on helpompi tulkita. Alla sijaitsevassa taulukko sisältää automaatiotiedon asettamisen määritykset (ks. Taulukko 3).

Taulukko 3. Automaatiotiedon asettamisen määritykset.

sliderAttach	Yhteys prosessorin ja säätimen välille
gainRange	Arvoalue uudelleen
createAndAddParameter	Parametrien konfigurointi



Kuva 10. Esimerkki FL Studio -ohjelmistosta, jossa automaatio ei ole kyseisellä hetkellä käytettävissä ääniliitännäisessä.

2.3.6 Edellisen arvon tallentaminen ohjelman muistiin

Äänenvoimakkuutta säätelevän ohjelman parantamiseksi on myös rakennettava toiminto, joka tallentaa edellisen arvon ohjelman muistiin. Tämä tarkoittaa siis sitä, että ohjelma muistaa sille asetetun arvon FL Studio -ohjelmiston projektin uudelleen aukaistaessa sen tallentamisen jälkeen. Ilman tätä toimintoa ohjelma ei muista sille aikaisemmin asetettua arvoa ja aukaistaessa FL Studio -ohjelmiston se palauttaa sille sen oletusarvon.

Tämä onnistuu hyödyntämällä XML-tiedostoa. Ensimmäisenä ohjelma lukee sen hetkisen tiedon ja luo XML-tiedoston. Tämän jälkeen tallentaessa projektin FL Studio -ohjelmistossa ohjelma asettaa sen hetkisen tiedon XML-tiedostoon. Ohjelmaan on myös rakennettu virheentarkistus.

Koodien positiot eri tiedostoissa on nähtävissä liitteissä (ks. Liite 3), joista ohjelmaan lisättyjä kyseisiä koodeja on helpompi tulkita. Alla sijaitseva taulukko sisältää edellisen arvon tallentamisen määrityksiä.

Taulukko 4. Edellisen arvon tallentamisen määritykset.

savedParams	Olion alustus
xml	Ohjelman sen hetkisen tiedon lukeminen ja XML-tiedoston luonti
theParams	Ohjelman sen hetkisen tiedon asettaminen XML-tiedostoon
theParams != nullptr	Lauseen määrittäminen virheentarkistusta varten

3 FL STUDIO 12

FL Studio (aiemmin tunnettu nimellä FruityLoops) on yksi maailman kuuluisimmista DAW (Digital Audio Workstation) –ääniohjelmistoista, joka on osoittautunut monien ammattimuusikoiden käyttämäksi sekä suosimaksi ohjelmistoksi. (Image-Line n.d-a.)

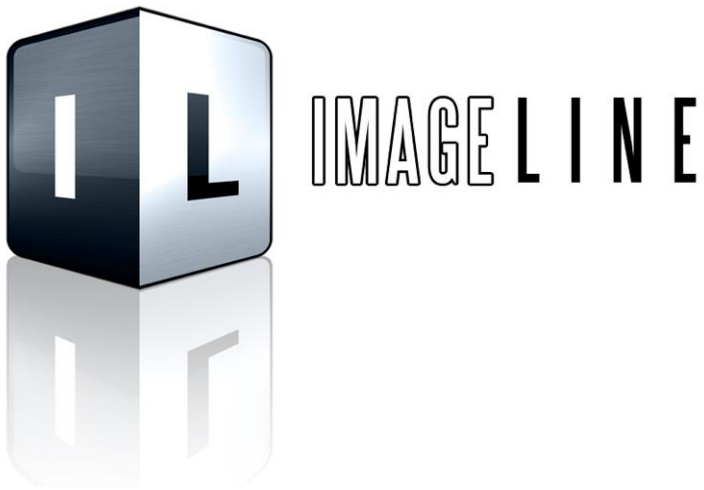
3.1 Historia

FL Studio -ohjelmistolle on kertynyt historiaa vuodesta 1994, ja sen kehityskaari on ollut nouseva tähänkin päivään saakka.

3.1.1 Image-Line

FL Studio sai alkunsa belgialaisen Image-Line yhtiön (ks. Kuva 11) toimesta, jonka perustivat Jean-Marie Cannie ja Frank Van Biesen vuonna 1994. Aluksi kyseinen kaksikko kehitti videopeliä, joka muistutti lähes identtisesti Tetristä. Videopelinsä julkaisun jälkeen he aloittivat työskentelyn suuren videopeliyhtiön Privaten kanssa. (Image Line n.d-b.)

Privatessa työskennellessään he kehittivät muutaman erittäin menestyneen pelin, ja tämän jälkeen yhtiö muutti nimensä Image-Lineksi vuonna 1994. Image-Line kiinnostui samoihin aikoihin henkilöstä nimeltä Didier Dambrin. Nuori 19-vuotias tietokonepelien kehittäjä oli aikaisemmin voittanut ”Da Vinci”-suunnittelijakilpailun, jonka oli järjestänyt IBM. Image-Line palkkasi lopulta Dambrinin tietokonepelien suunnittelijaksi. (Image Line n.d-b.)



Kuva 11. Image-Linen logo.

3.1.2 FruityLoops

Hetkiä myöhemmin Dambrinilla heräsi mielenkiinto musiikkisovellusten kehittämisen puolelle. Hän aloitti kehittää sovellusta nimeltä FruityLoops. Aluksi tämä oli ainoastaan kaikessa yksinkertaisuudessaan MIDI-rumpukone (ks. Kuva 12). Image-Line ei ollut aluksi varma, kuinka hyvin ohjelma saa suosiota, mutta tämä paljastui lopulta turhaksi huolenaiheeksi. FruityLoops julkaistiin virallisesti vuonna 1998, ja ohjelma sai uskomattoman suosion. (Image Line n.d-b.)

Kuva 12. FruityLoops-sovelluksen ensimmäinen versio.

3.1.3 FruityLoops FL Studio -ohjelmistoksi

Noin neljän vuoden kehittämisen jälkeen nimi vaihdettiin, ja sitä alettiin kutsua nimellä FL Studio. Nimen vaihtoon oli muutamia syitä. Kelloggs päätti haasta heidät oikeuteen, koska FruityLoops muistutti erittäin paljon erästä heidän tuotemerkkiään (ks. Kuva 13). Kustannuksellisista syistä nimi päätettiin muuttaa. Nimen vaihtoon vaikutti myös se, että sanasta ”Loops” voi saada käsityksen, että ohjelmalla rakennettu sisältö perustuu vain ja ainoastaan valmiisiin ääniin, jotka sisältävät esimerkiksi rumpurytmejä. Myöskään sana ”Fruity” ei kuulostanut hyvältä markkinoinnin kannalta. (Image Line n.d-b.)

Kuva 13. FL Studio -ohjelmiston logo.

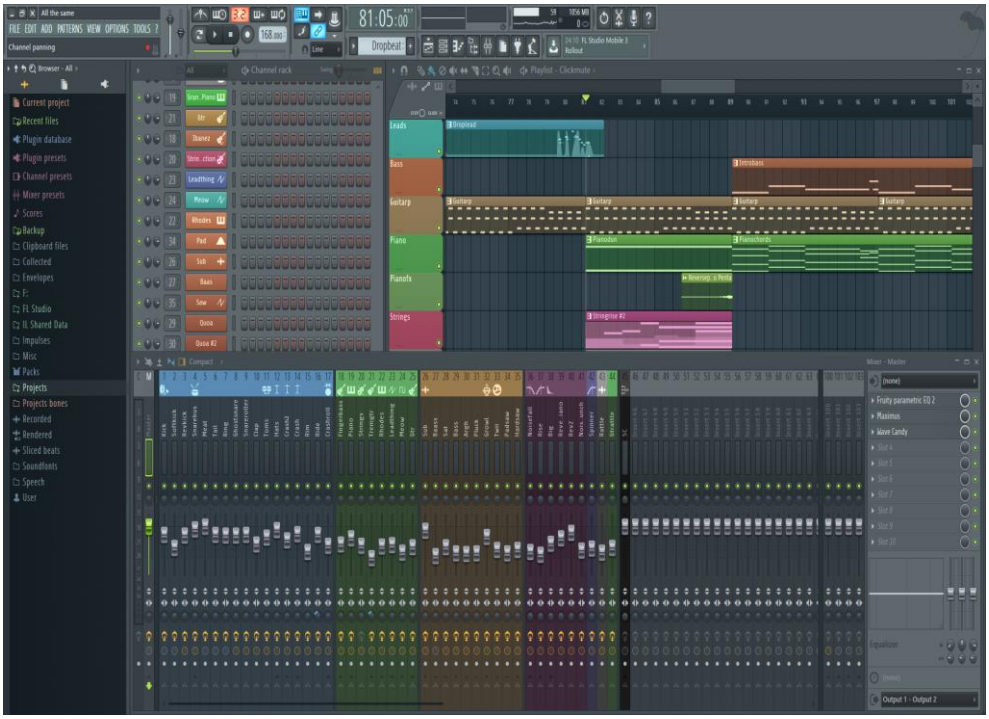
3.2 Käyttövaatimukset

2Ghz Intel Pentium 4 / AMD Athlon 64 (tai uudemman) prosessorin täydellä SSE2 tuella, 32- tai 64-bittisen Windows 10/8.0/8.1/7 tai Intel Mac johon on asennettu Boot Camp / Windows tai Intel Mac OS X 10.8 tai 10.9 FL Studio Mac OS X beta-versiolle. 1 GB tai enemmän keskusmuistia (RAM), 1 GB vapaata kovalevytilaa ja äänikortti DirectSound-ajureilla. ASIO/ASIO2 yhteensopivuus vaaditaan äänen nauhoittamista varten.

On hyvä muistaa, että kyseiset vaatimukset ovat vain minimivaatimuksia. Mitä enemmän tehoa työasemasta löytyy, sitä paremmin ohjelmisto jaksaa suorittaa raskaampiakin projekteja. (Image-Line n.d-d.)

3.3 Käyttöliittymä ja sen käytettävyys

Seuraavana aiheena on FL Studio -ohjelmiston käyttöliittymä, joka sisältää monia hyödyllisiä painikkeita ja valikoita käyttäjän tueksi. Alla sijaitsevassa kuvassa (ks. Kuva 14) on havainnollistettu FL Studio -ohjelmiston esimerkkiprojekti. (Image-Line n.d-c.)



Kuva 14. FL Studio sisältää muutamien artistien tekemiä demokappaleprojekteja, joista tässä on esimerkkinä yksi niistä.

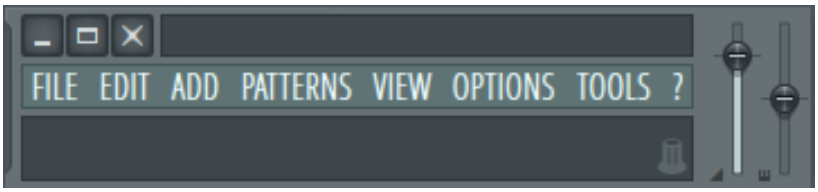
3.3.1 Päävalikko

Ohjelmiston vasemmassa yläkulmassa löytyy päävalikko (*engl. Main Bar*) (ks. Kuva 15). ”File”-valikosta käyttäjä voi luoda uuden projektin, avata jonkin vanhoista projekteista, tallentaa projektin, tuoda MIDI-tiedoston ja

viedä ääniraidan käyttäjän haluamassa muodossa. Viimeisenä näkyvissä on myös viimeisimmät projektit, joita käyttäjä on käsitellyt.

”Edit”-valikosta käyttäjä voi peruuttaa viimeisimmän tekemänsä muutoksen, kopioida, liittää sekä leikata. ”Add”-valikosta käyttäjä voi lisätä instrumentteja sekä efektejä. ”View”-valikosta käyttäjä voi muokata näkymiä piiloon sekä näkyviin. ”Options”-valikosta käyttäjä löytää MIDI-, ääni-, pää- ja tiedostoasetukset, projektin asetukset, projektin tiedot sekä projektin muutoslokin.

”Tools”-valikosta käyttäjä voi päättää, ottaako hän käyttöön ohjelmia, kuten esimerkiksi äänityksen aloittamisen yhdellä napin painalluksella. Kysymysmerkistä eli ”Help”-valikosta löytyy apua eri aiheista. Päävalikosta löytyy vielä äänenvoimakkuuden (*engl. Main Volume*) sekä äänenkorkeuden (*engl. Main Pitch*) säätimet.



Kuva 15. FL Studio -ohjelmiston päävalikko.

3.3.2 Paneelit

Päävalikon oikealta puolelta löytyy paneelit (*engl. Panels*) (ks. Kuva 16), jotka helpottavat työskentelyä ja näyttävät myös suorituskyvyllistä tietoa. Kuvan vasemmalta puolelta löytyy ”Play”-, ”Stop”- ja ”Record”-painikkeet. Siitä löytyy myös metronomi, liukusäädin temmon muokkaamiselle sekä valitsin ”Song”- ja ”Pattern”-tilan välillä. ”Pattern”-tilassa rakennettaan yksittäisistä tahtipalkeista yhtä projektin osaa, ja ”Song”-tilassa rakennetaan projektin kokonaiskuvaa.

Seuraavaksi on aikapaneeli, joka näyttää ajan minuutteina ja sekunteina tai myös palkkeina. Aikapaneelin vierestä löytyy äänen ulostulon seuraamiseen monitori, josta voi seurata prosessorin käyttäytymistä sekä muistin käyttöä. Näiden alta löytyy pikanäppäimiä, joilla käyttäjä saa näkyviin sekä piilotettua ”Playlist”-, ”Channel Rack”-, ”Piano Roll”-, ”Browser”- ja ”Mixer”-näkymän.

Kuva 16. FL Studio -ohjelmiston paneelit.

3.3.3 Mikseri

FL Studio -ohjelmisto sisältää mikserin (*engl. Mixer*) (ks. Kuva 17), jonka lävitse kaikki ääni kulkee. Mikserissä on yli sata (103) kanavaa (*engl. Mixer Tracks*), ja jokaiselle kanavalle on mahdollista sisällyttää erilaisia instrumentteja, äänitiedostoja tai efektejä. Jokaiselle kanavalle on mahdollista vaihtaa värin ja kuvakkeen, jotta mikserin selailu olisi helpompaa monen kanavan ollessa käytössä. Näiden lisäksi mikserissä on yksi pääkanava (*engl. Master Track*), joka hallitsee ensisijaisesti jokaista kanavaa. Jokaisen kanavan kohdalla on oma voimakkuusmittarinsa *(engl. Level Fader*), josta käyttäjä voi seurata kanavan laatua. Mikserissä on mahdollista reitittää (*engl. Send Switch*) yksi tai useampi kanava toiselle kanavalle kuten alla olevassa kuvassa ”Clap”-kanava on reititetty kanavaan ”Insert 5”. Tämä mahdollistaa efektien tehostamisen paljon laajemmin.

Jokaisella kanavalla on oma äänenvoimakkuuden liukusäätimensä ja pienen, vihreän pallon kohdalta käyttäjä voi mykistää (*engl. Mute*) kanavan täysin tai päinvastoin. Monien kanavien ollessa käytössä tämä on hyvä tapa seurata ja työskennellä eri kanavien parissa. Jokaisella kanavalla on myös säädin stereoäänen kiertämiselle (*engl. Panning*), joka sijaitsee mykistyspainikkeen alapuolella. Säädin kiertää stereoääntä joko vasemmalle tai oikealle puolelle. Äänenvoimakkuuden liukusäätimen alapuolelta löytyy myös säädin stereoäänen erottamiselle (*engl. Stereo Separation*), joka mahdollistaa olemassa olevien stereoäänien tehostamisen tai laskemisen kanavalla.

Kuvan oikeassa sivussa on kymmenen efektipaikkaa, johon käyttäjä voi asettaa haluamiansa tehosteita kanavalle. Esimerkkinä alla olevassa kuvassa on asetettu kaksi efektiä, tehosteissa käyttäjä voi myöskin halutessaan hiljentää ne ja sekä päinvastoin. Myöskin tehosteen äänenvoimakkuutta on mahdollista säädellä tehosteen oikealla puolella sijaitsevasta säätimestä. Efektien alapuolella on oma taajuuskorjaimensa (*engl. Equalizer*), josta voidaan säädellä ja korjailla kanavan taajuuksia. Oikeasta yläreunasta löytyy äänikortin sisääntulon (*engl. Input*) valitsin, sekä alareunasta löytyy äänikortin ulostulon (*engl. Output*) valitsin.

Kuva 17. FL Studio -ohjelmiston mikseri.

3.3.4 Soittolista

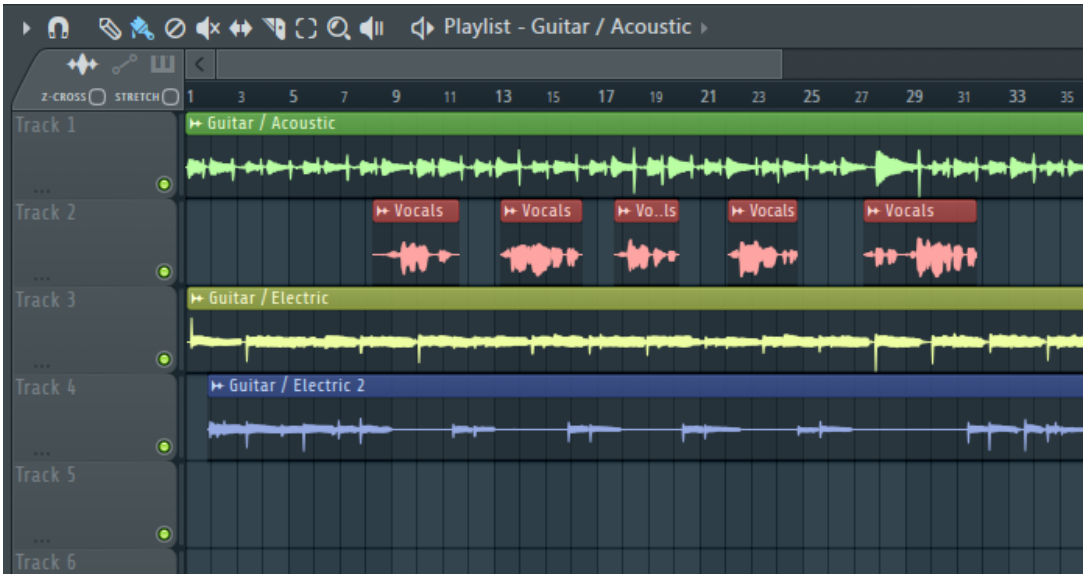
FL Studio -ohjelmiston soittolista (*engl. Playlist*) on ohjelmiston keskeisin työkalu, jolla musiikkia tai ääniprojektia rakennetaan. Soittolistalle syötetään erilaisia äänitiedostoja tai eri instrumenttien tahtipalkkeja. Alla olevassa kuvassa on havainnollistettu soittolistan toimintaa muutamalla nauhoitetulla kitaran äänitiedostolla sekä yhdellä nauhoitetulla laulun äänitiedostolla (ks. Kuva 18). Äänitiedostot sijoitetaan eri raidoille ja niistä rakennetaan yksi iso kokonaisuus.

Äänitiedostojen sijoittaminen on erittäin yksinkertaista, käyttäjä voi halutessaan hakea työaseman tiedoston tiedostosijainnista ja raahata soittolistalle. Käyttäjä voi siirrellä äänitiedostoja haluamalleen kohdalle yksinkertaisesti hiirtä käyttämällä. Alla olevassa kuvassa on myös selkeytetty, että raitoja on mahdollista nimetä haluamansa mukaan ja antaa niille värejä sekä kuvakkeita.

Vihreä pallo toimii edellisten mainintojen tapaan samalla tavalla mykistämisen painikkeena. Soittolistan vasemmasta yläkulmasta käyttäjä saa esiin soittolistan asetukset. Instrumenttien palkkikuvioiden lisääminen tapahtuu piirtämällä (*engl. Draw*) asettamalla ne yksi kerrallaan soittolistalle tai maalamalla (*engl. Paint*) sijoittamalla ne hiirtä vetämällä soittolistalle.

Yläpalkista löytyy myös painikkeet, raidan poistamiselle *(engl. Delete*), mykistämiselle, raidan siirtelylle aikajanalla (*engl. Slip*), raidan silpomiselle käyttäjän haluamista kohdista (*engl. Slice*), monien raitojen samanaikai-

seen valitsemiseen (engl. Select) ja siirtämällä aikajanaa käyttäjän haluamalle kohdalle pienemmäksi tai suuremmaksi (*engl. Zoom*). Soittolistan painikkeiden alapuolelta on mahdollista myös valita, keskittyykö soittolista äänitiedostoihin vai instrumenttien palkkikuvioiden kanssa työskentelyyn.



Kuva 18. FL Studio -ohjelmiston soittolista, johon on esimerkkinä nauhoitettu muutama ääniraita.

3.3.5 Kanavaikkuna ja askelsekvensseri

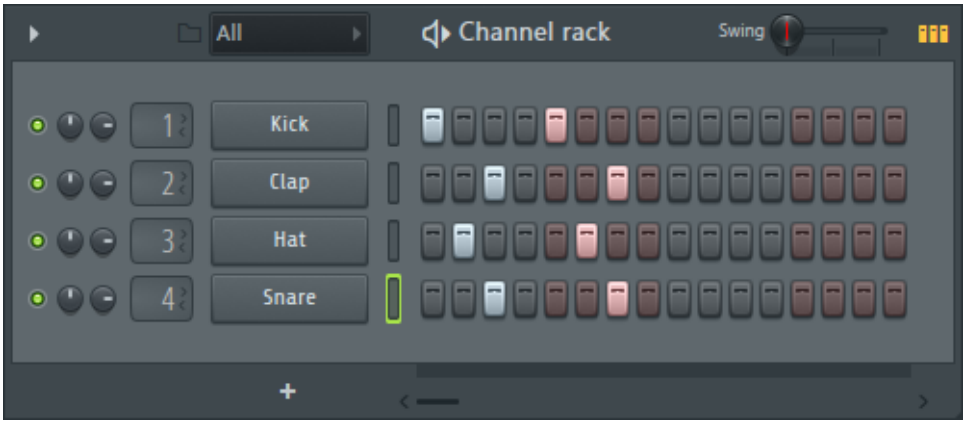
FL Studio- ohjelmiston kanavaikkuna (*engl. Channel Rack*) sisältää oman askelsekvensserin (*engl. Step Sequenser*). Kanavaikkunan eri kanaville syötetään instrumentteja, jonka jälkeen muodostetaan omia sävellyksiä sekvensserin avulla.

Sekvensserillä luodusta palkkikuvioinnista (*engl. Pattern*) rakentuu melodia, joka määrittyy instrumentin ja erikseen määritellyn tahdin ja temmon mukana. FL Studio- ohjelmistossa palkkikuviointeja on mahdollista rakentaa erilaisia, ja tyhjiä pohjia palkkikuvioinnille on ohjelmistossa satoja. Kanavaikkunasta voidaan siirtää eri instrumenttien palkkikuviointeja soittolistalle.

Alla sijaitsevassa kuvassa on havainnollistettu kanavaikkunan ja askelsekvensserin toimintaa (ks. Kuva 19). Palkkeja voidaan lisätä sekvensseriin hiiren vasemmalla painikkeella ja halutessa poistaa hiiren oikealla painikkeella. Instrumentteja voidaan siirtää kanavaikkunaan FL Studio- ohjelmiston tiedostoselaimesta tai alakulmassa sijaitsevasta ”+”-painikkeesta.

Kanavaikkunaan on alla olevassa esimerkissä syötetty eri rumpuääniä. Instrumentin oikealla puolella sijaitseva vihreä valo kertoo, mikä instrumentti on sillä hetkellä käytössä. Instrumentin vieressä sijaitseva numero kertoo, mille mikserin kanavapaikalle instrumentti on sijoitettu.

Kanavaikkunan vasemmassa reunassa sijaitsee painike mykistämiselle ja säätimet stereoäänen kiertämiselle ja kanavan instrumentin äänenvoimakkuudelle. Vasemmasta yläkulmasta löytyy myös kanavaikkunan asetukset.



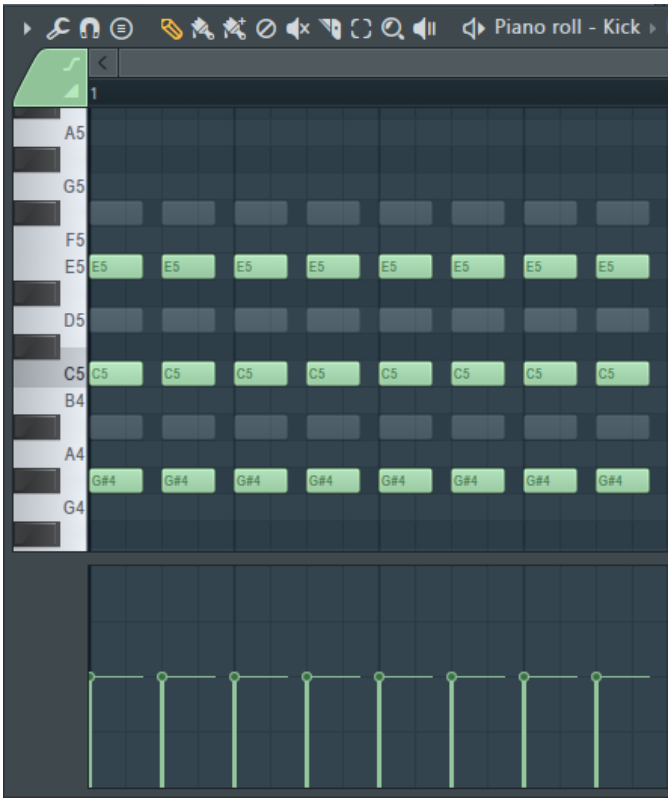
Kuva 19. FL Studio -ohjelmiston kanavaikkuna ja askelsekvensseri.

3.3.6 Piano Roll -työkalu

FL Studio -ohjelmiston Piano Roll -työkalun tarkoituksena on lähettää nuotteja sekä automaatiotietoa instrumenteille, jotka on asetettu työkalun kanaville. Alla sijaitsevassa kuvassa on havainnollistettu, kuinka työkalu toimii (ks. Kuva 20), jossa rumpuinstrumentille on syötetty nuotteja.

Käyttäjä syöttää nuotteja haluamilleen sävelkorkeuksille, joista projektin pohja rakentuu. Kuvan yläreunassa löytyy samoja asetuksia ja pikatyökaluja, joita on selvennetty jo aikaisemmin.

Selkeämmin näkyvät nuottipalkit ovat sen instrumentin nuotteja, joiden kanssa käyttäjä on kyseisellä hetkellä työskentelemässä ja taas harmaalla, hieman heikommin näkyvät palkit, ovat toisen instrumentin nuotteja, joita kanavalle on asetettu. Jokaista nuottia on mahdollista muokata erikseen käyttäjän haluamalla tavalla työskentelyn helpottamiseksi.



Kuva 20. FL Studio -ohjelmiston Piano Roll -työkalu.

Piano Roll -työkalua on mahdollista ohjata myös kanavaikkunasta, alla sijaitsevassa kuvassa on havainnollistettu miltä työkalu näyttää kanavaikkunassa (ks. Kuva 21). Käyttäjän luoma melodia instrumenteille näkyy kanavaikkunassa sekvensserin paikalla.

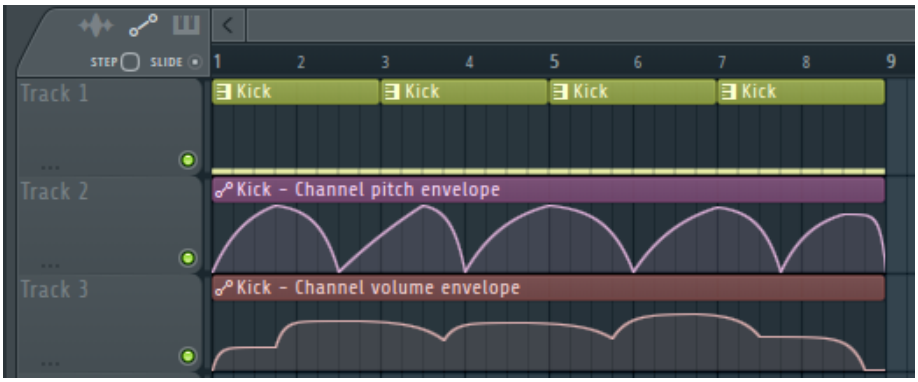
Kuva 21. FL Studio -ohjelmiston Piano Roll -työkalu kanavaikkunan näkymässä.

3.3.7 Automatisointi

FL Studio -ohjelmisto sisältää työkalun, jolla voi automatisoida erilaisia toimintoja toimimaan soittolistalla. Esimerkkinä, alla olevassa kuvassa on automatisoitu tavalliselle ”Kick”-rumpuinstrumentille äänenvoimakkuuden sekä äänen sävelkorkeuden automaattinen muuttuminen ääniprojektissa (ks. Kuva 22). Automatisointi tapahtuu seuraavalla tavalla.

Käyttäjä valitsee instrumentin asetuksista jonkin säätimen, kuten esimerkiksi äänenvoimakkuuden. Yksinkertaisesti käyttäjä säätelee ja heiluttelee säädintä niin laajalla skaalalla, kuin tahtoo automatisoinnin sitä käsittele-

vän. Tämän jälkeen käyttäjä käy luomassa automatisoidun raidan ohjelman asetuksista (Tools – Last Tweaked – Create Automation Clip). Raita ilmestyy kanavaikkunaan ja sen voi lisätä soittolistalle. Soittolistalla käyttäjä voi ohjata raitaa käyttäytymään haluamallansa tavalla. Esimerkkinä, äänenvoimakkuuden automatisoinnissa ääniviivan ollessa matalimmalla, ääni on mykistetty, kun taas ääniviivan ollessa korkeimmalla, se toistaa äänen kovimmalla mahdollisella asteella.

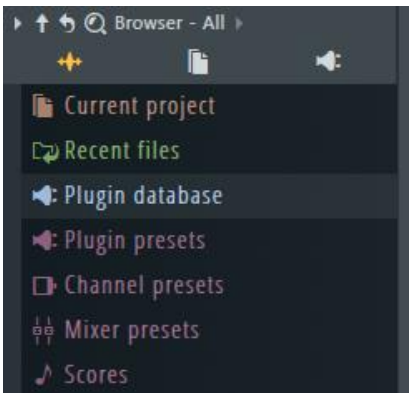


Kuva 22. ”Kick”-rumpuinstrumentille automatisoitu äänenvoimakkuuden ja äänen sävelkorkeuden säätely FL Studio -ohjelmistossa.

3.3.8 Tiedostoselain

FL Studio -ohjelmiston tiedostoselaimen (*engl. File Browser*) kautta käyttäjä voi selata projektin tietoja. Alla sijaitsevassa kuvassa on havainnollistettu tiedostoselaimen näkymä (ks. Kuva 23).

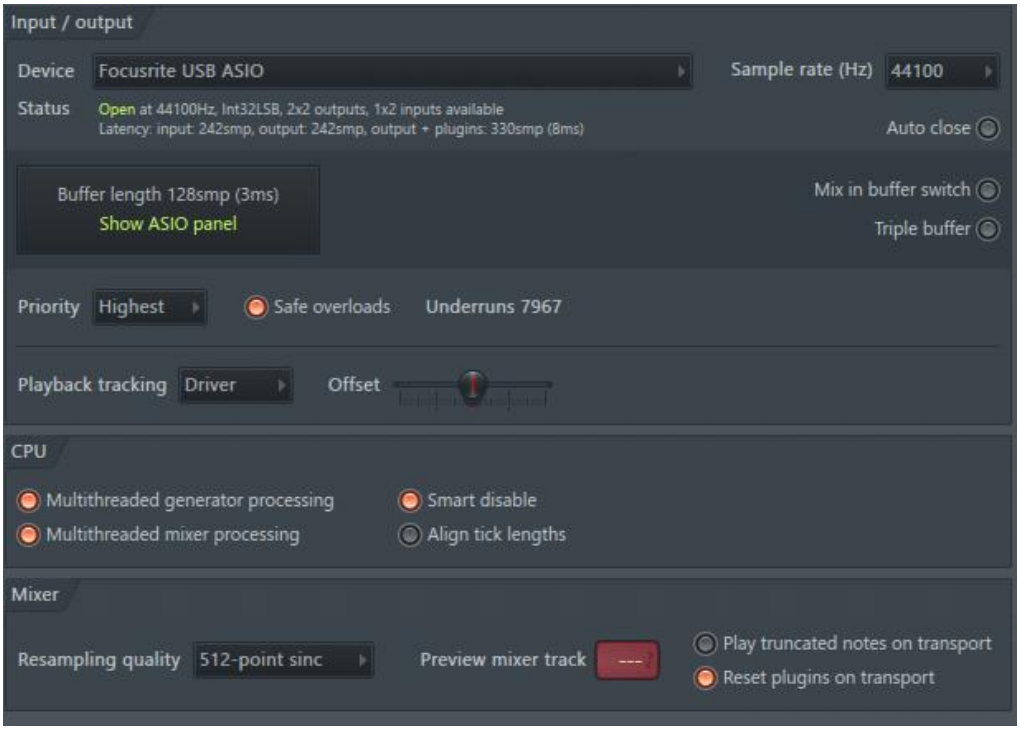
Käyttäjä voi selata sieltä efektejä, instrumentteja, äänitiedostoja, nykyistä projektia, aikaisemmin käytettyjä tiedostoja, efektien ja instrumenttien tietokantaa, efekteille rakennettuja pohjia ja esiasetuksia (*engl. Presets*), kanaville rakennettuja pohjia ja esiasetuksia, mikserille rakennettuja pohjia ja esiasetuksia, valmiiksi rakennettuja automaatioita melodioille sekä eri artistien kehittämiä demokappaleita ohjelmistolle. Kaikki tiedostoselaimesta löytyvä on mahdollista siirtää selaimesta suoraan aikajanalle, mikseriin, kanavaikkunaan tai Piano Roll -työkaluun.



Kuva 23. FL Studio -ohjelmiston tiedostoselain.

3.3.9 Äänikortti

Erittäin olennainen osa ääni- ja musiikkiohjelmistoa on tehokas äänikortti. MIDI-kontrollerin ja mikrofonin yhdistäminen äänikorttiin on myös helpottava tekijä ääniprojekteja varten. FL Studio vaatii ASIO4ALL-ajurin toimiakseen, joka löytyy heidän verkkosivuiltaan tai äänikortin valmistajan kotisivuilta. Alla olevassa kuvassa näkyy äänikortin asetukset (ks. Kuva 24). Laitteen kohdalta avautuu vetovalikko, josta käyttäjän on valittava äänikorttinsa.



Kuva 24. Äänikortin asetukset FL Studio -ohjelmistossa.

3.4 **Liitännäiset**

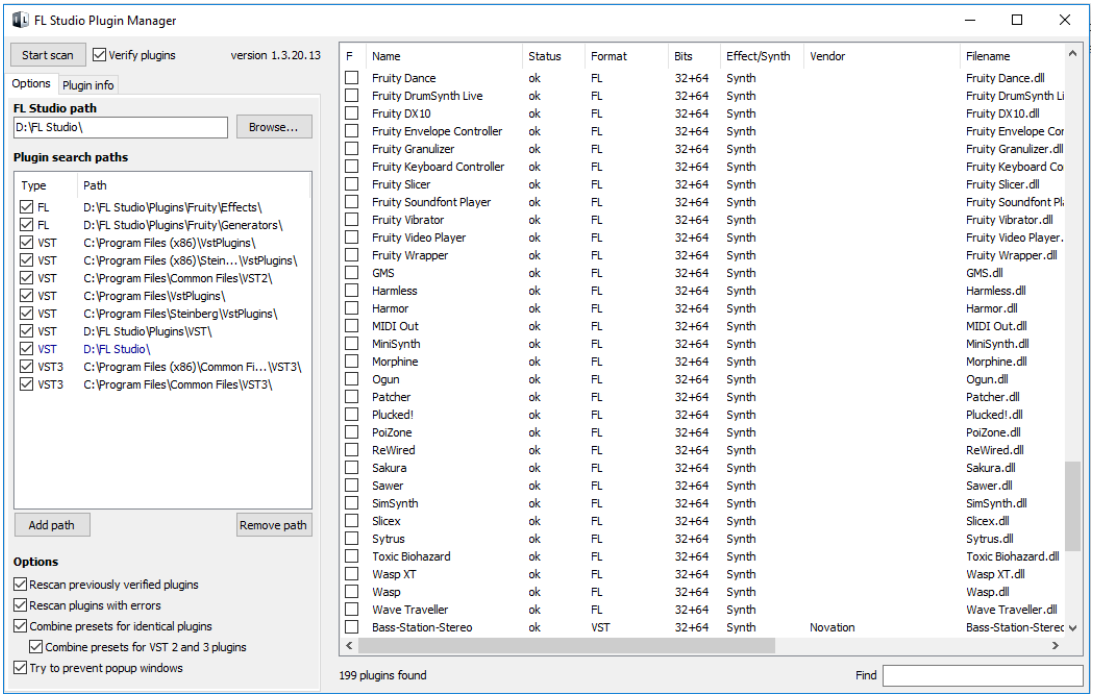
FL Studio -ohjelmiston liitännäiset ovat musiikinteon ja ääniprojektien yksi olennaisimmista tekijöistä. Niitä kutsutaan myös nimellä VST (Virtual Studio Technology) -plugin. Liitännäinen on tietokoneohjelma, joita käytetään FL Studio -ohjelmiston tapaisten ohjelmistojen kanssa.

FL Studio sisältää paljon erittäin hyödyllisiä liitännäisiä, joita löytyy verkosta lisää niin ilmaisena kuin maksullisenakin. Liitännäisiä lukeutuu muutamiin erilaisiin aihealueisiin, joita on listattu tämän luvun seuraavissa vaiheissa. FL Studio -ohjelmistossa on paljon liitännäisiä, mutta työssä on käsitelty niistä vain muutamia.

3.4.1 VST-liitännäisten asentaminen

Liitännäisten asentaminen FL Studio -ohjelmistoon on erittäin yksinkertaista. DLL-tiedostomuotoinen liitännäinen siirretään ohjelmiston asentamisen yhteydessä luotuun VST-kansioon, joka löytyy ohjelmiston tiedostopolusta (FL Studio – Plugins – VST), minkä jälkeen ne on lisättävä ohjelmistoon sen asetuksista.

Asetuksista avautuu ”Plugin Manager”, josta työasemassa sijaitsevia liitännäistiedostoja hallinnoidaan. Kuvan vasemmassa reunassa on hakemisto, johon on määritelty kansiot, josta ohjelma etsii liitännäisiä. Etsimisen jälkeen liitännäinen siirtyy listalle ja on nyt FL Studio -ohjelmiston käytettävissä (ks. Kuva 25).



Kuva 25. FL Studio -ohjelmiston ”Plugin Manager”.

3.4.2 Fruity Compressor

Fruity Compressor -liitännäinen (ks. Kuva 26) on äänen kompressointiin perustuva efekti. Kompressointi tarkoittaa äänisignaalin voimakkuuden pienentämistä ja kompressoinnilla tavoitellaan äänenvoimakkuuden kasvattamista ilman, että kaikkein suurin äänenvoimakkuus kasvaa. Toisin sanoen, kompressoinnilla saadaan ääni kuulostamaan kovemmalta ja äänisignaali ei nouse liian korkealle ja projektia häiritsevälle tasolle.

Kompressori sisältää säätimen äänenvoimakkuuden ylittämisen kynnysarvolle *(engl. Threshold)*, jonka se laskee annettuun arvoon asetetun suhteen *(engl. Ratio)* perusteella. Muokattavissa on myös aika, kuinka kauan kes-

tää, että vaimennus iskeytyy päälle (*engl. Attack*) ja aika, kuinka kauan kestää, että vaimennus loppuu (*engl. Release*). Säädettävissä on myös kompressoinnin jyrkkyys *(engl. Knee)*, eli millä tavalla kompressointi alkaa kynnysarvon ylitettyä, joko jyrkästi *(engl. Hard)* tai loivasti *(engl. Soft)*.

Kuva 26. Fruity Compressor -liitännäinen.

3.4.3 Fruity Limiter

Fruity Limiter -liitännäinen (ks. Kuva 27) on myös äänen kompressointiin perustuva efekti, jonka tarkoituksena on rajoittaa äänisignaalia niin, ettei se ylitä erikseen asetettua rajaa. Tällä estetään liian kovat äänenvoimakkuudet sekä ylimääräistä säröisyyttä ja kohinaa. Liitännäinen sisältää rajoittimen sekä kompressorin.

Kuva 27. Fruity Limiter -liitännäinen.

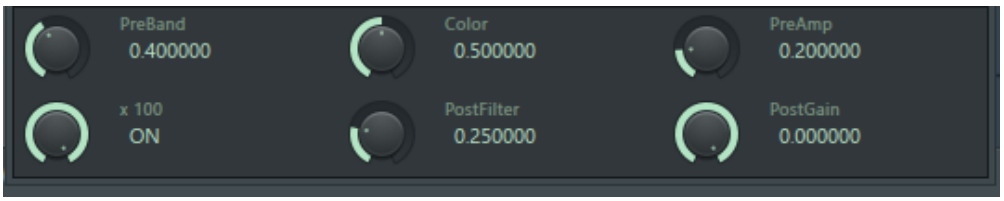
3.4.4 Fruity Delay 2

Fruity Delay 2 -liitännäinen (ks. Kuva 28) on viive-efekti (*engl. Delay*), jolla saadaan ääniraita soimaan uudelleen ja uudelleen efektin asetuksien määrittelemän ajan mukaan. Liitännäinen sisältää erilaisia säätimiä, joista käyttäjä voi säädellä viive-efektin voimakkuutta, kestoa ja muotoa.

Kuva 28. Fruity Delay 2 -liitännäinen.

3.4.5 Fruity Blood Overdrive

Fruity Blood Overdrive -liitännäinen (ks. Kuva 29) on säröefekti, jonka tarkoituksena on luoda ääneen säröä. Tämä on erittäin hyödyllinen efekti niin sähkökitaraa, kuin bassoakin sisältävissä äänitiedostoissa. Säröefekti lisää äänenvoimakkuutta, tekee äänestä kompressoidumman ja rikkoo ääntä tarkoituksellisesti. Liitännäisessä käyttäjä voi säädellä säröefektin voimakkuutta ja tehdä siitä myös pehmeämmän kuuloisen.



Kuva 29. Fruity Blood Overdrive -liitännäinen.

3.4.6 Fruity Parametric EQ 2

Fruity Parametric EQ 2 -liitännäinen (ks. Kuva 30) on taajuuksien korjaamiseen ja säätelyyn perustuva efekti, jonka tarkoituksena on nostaa tai laskea äänenvoimakkuutta erikseen määritellyillä taajuuksilla.

Alla sijaitsevassa kuvassa on havainnollistettu efektin toimintatapaa. Kuvassa näkyvät numeroidut ympyrät määrittävät äänen kulun eri taajuuksilla. Käyttäjä voi siirrellä ympyröitä eri taajuusalueille haluamansa mukaan yksinkertaisesti hiiren avulla, sekä hiiren oikeasta painikkeesta käyttäjä voi muokata jokaista taajuusalueen ympyrää. Taajuuksia on mahdollista säädellä myös kuvan oikealla puolella näkyvistä liukusäätimistä.

Jokaiselle taajuusympyrälle on mahdollista antaa automatisoituja asetuksia, jotta taajuuskorjain toimisi tietyllä ja halutulla tavalla.

Kuva 30. Fruity Parametric EQ 2 -liitännäinen.

3.4.7 Fruity Love Philter

Fruity Love Philter -liitännäinen (ks. Kuva 31) on suodatinefekti (*engl. Filter*), joka sisältää kahdeksan samanlaista yksikköä. Yksiköt on mahdollista yhdistää toimimaan peräkkäin ohjautumalla seuraavaan jonossa olevaan yksikköön, tai sitten yksi kerrallaan erikseen. Jokaisessa yksikössä käyttäjä voi korjata äänentaajuuksia sekä valittavissa on valmiita suodattimen tyyppejä (*engl. Filter Type*), jotka säätävät taajuuksia automaattisesti toimimaan tietyllä taajuusalueella.

Suodattimen tyyppejä voidaan kytkeä joko kokonaan pois päältä, tai sitten tehostaa kaksin- tai kolminkertaiseksi. Taajuuksia on mahdollista myös säädellä käyttäjän haluamalla tavalla. Liitännäinen sisältää erilaisia esiasetuksia, joista käyttäjä voi valita haluamansa mukaan. FL Studio -ohjelmiston valmistajien mukaan kyseessä on monipuolisin suodatinefektien luomiseen kehitetty liitännäinen koko ohjelmassa.

Kuva 31. Fruity Love Philter -liitännäinen.

3.4.8 Fruity Chorus

Fruity Chorus -liitännäinen (ks. Kuva 32) on kuoroefekti, jonka tarkoituksena on saada ääniaalto soimaan kahteen tai useampaan kertaan samalla tavalla, mutta eri äänentaajuuksilla ja -korkeuksilla. Käyttäjä voi halutessaan muokata liitännäisessä kuoroefektin viivettä, syvyyttä sekä taajuuksia. Liitännäinen sisältää muutamia esiasetuksia, joista käyttäjä voi valita haluamansa.

Kuva 32. Fruity Chorus -liitännäinen.

3.4.9 Fruity Reeverb 2

Fruity Reeverb 2 -liitännäinen (ks. Kuva 33) on kaikuefekti, jonka tarkoituksena on muokata äänen akustiikkaa ja antaa sille erilaista tilan tuntua. Esimerkkinä, kun ihminen taputtaa käsiänsä yhteen erilaisessa tilassa, kuten kylpyhuoneessa tai konserttihallissa, syntyy erilainen ääni. Kuvan vasemmassa reunassa havainnollistetaan tilan kokoa.

Muokkaamalla liitännäisen säätimiä, kaiku muuttuu erilaiseksi. Liitännäisessä on mahdollista säätää bassoa, tilan kokoa, kaikuefektin kestoa ja kaikuefektin voimakkuutta. Liitännäisestä löytyy säädin myös stereoäänen siirtämiselle vasemmalle tai oikealle puolelle. Liitännäinen sisältää erilaisia esiasetuksia, joista käyttäjä voi valita haluamansa.

Kuva 33. Reeverb 2 -liitännäinen.

3.4.10 Fruity Stereo Enhancer

Fruity Stereo Enhancer -liitännäinen (ks. Kuva 34) on stereoääniin rakennettu apuväline. Liitännäisessä käyttäjä voi asettaa stereoäänen toimimaan joko vasemmalla tai oikealla puolella. Lisäksi liitännäisessä voi rakentaa efektin, joka soittaa toista puolta äänestä hieman viiveellä. Esimerkkinä alla olevassa kuvassa on havainnollistettu, että stereon oikeanpuoleinen kanava tulee viiveellä, kun säädin on säädetty oikealle puolelle säädintä.

Kuva 34. Fruity Stereo Enhancer -liitännäinen.

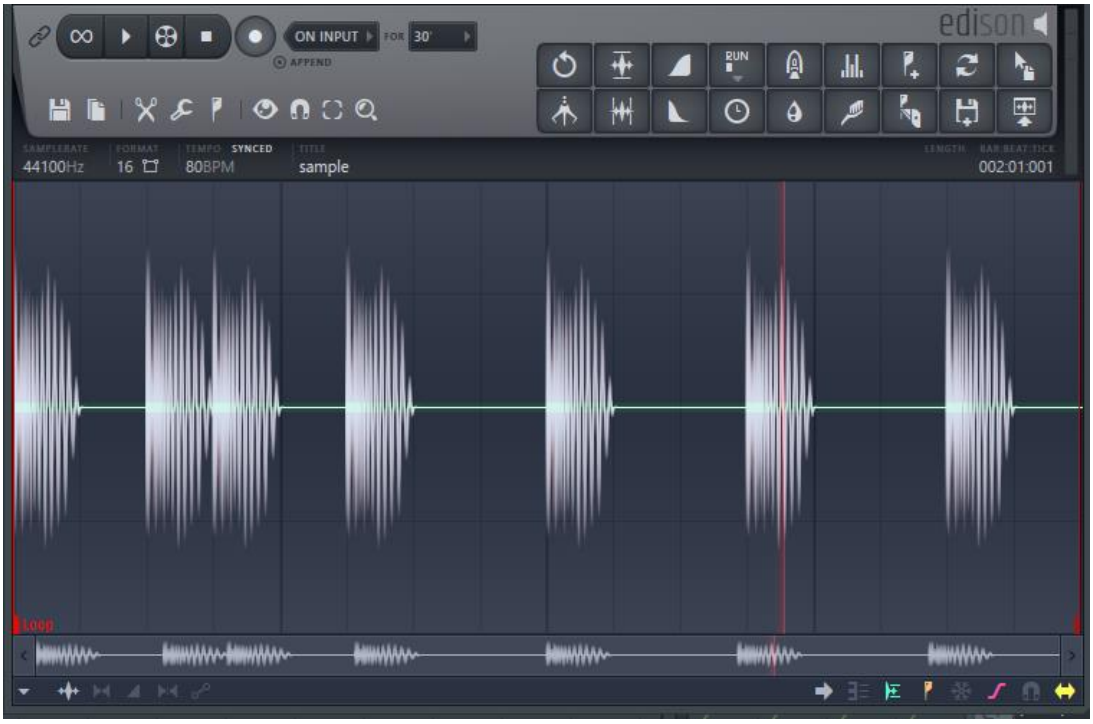
3.4.11 Edison

Edison -liitännäinen (ks. Kuva 35) on äänitiedoston nauhoittamiseen ja muokkaamiseen rakennettu työkalu. Edison on loistava työkalu äänitiedoston nauhoittamiseen, soittamiseen, tietojen muokkaamiseen ja leikkaamiseen.

Edison näyttää näytteenottotaajuuden (*engl. Sample Rate*), äänitiedoston bitit (*engl. Bit Depth*), temmon sekä äänitiedoston nimen. Käyttäjä voi myös muokata, haluaako äänitiedoston olevan monon- vai stereon muodossa (*engl. Channels*). Käyttäjä voi valita, näyttääkö liitännäinen temmon synkattuna projektin tempoon vai vapaassa muodossa. Edisonilla on myös erittäin helppoa rakentaa luuppeja, koska äänitiedoston saa soimaan loputtoman pitkään peräkkäin.

Edisonilla onnistuu asetusvalikon alta stereoäänen erottelu vasemmalle tai oikealle puolelle, äänitiedoston normalisointi (*engl. Normalize*), alku- (*engl. Fade In*) tai loppuhäivytys (*engl. Fade Out*), kohinan tai häiriöäänten etsiminen (*engl. Acquire Noise Profile*) äänitiedostosta, kohinan poisto käyttäjän haluamalta alueelta (*engl. Clean Up*), kaiun lisääminen, taajuuskorjaus sekä äänitiedoston lähettäminen soittolistalle (*engl. Send To Playlist As Audioclip*) ja äänitiedoston lähettäminen mikserin kanavalle (*engl. Send To Selected Channel*).

Edisonista käyttäjä voi tallentaa helposti äänitiedostoja työasemalleen. Käyttäjä voi lisätä äänitiedostoon merkintöjä, jotka helpottavat projektin rakentamisessa. Kuvan oikeassa yläkulmassa on lukuisia pikanäppäimiä liitännäisen eri toiminnoille.



Kuva 35. Edison -liitännäinen.

4 FL STUDIO JA SEN VERTAILU MUIHIN SITÄ VASTAAVIIN ÄÄNIOHJELMISTOIHIN

Tässä vaiheessa opinnäytetyötä on aiheena vertailu, jossa verrataan FL Studio -ohjelmistoa muihin vastaaviin ääniohjelmistoihin. Vertailun kohteena on kolme ohjelmistoa, ja ne ovat Adobe Audition, Ableton Live 9 Lite sekä Pro Tools. Osa ohjelmistoista vaatii lisenssin.

Työssä on pyritty havainnollistamaan ohjelmistojen toimivuutta, käyttöliittymää, helppokäyttöisyyttä ja työnjälkeä perin pohjin. Työssä on pyritty myös jakamaan mielipiteitä ohjelmista, jotka voivat luonnollisesti olla jokaisen ihmisen kohdalla erilaisia. Lähtökohtaisesti mikä tahansa kyseisistä ohjelmistoista voi olla käyttäjälle paras mahdollinen vaihtoehto.

4.1 Adobe Audition

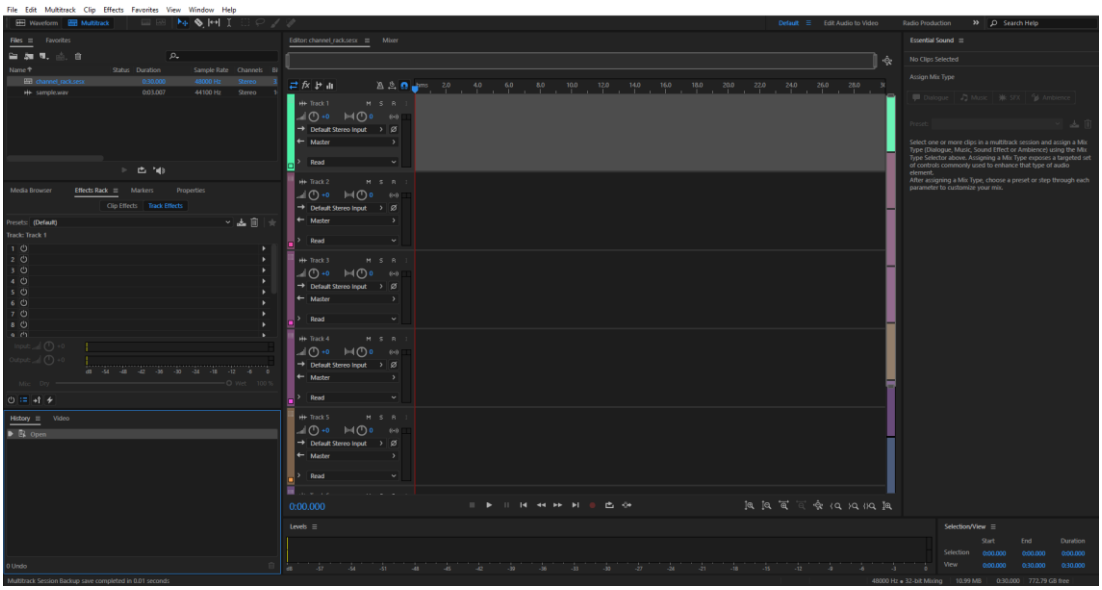
Adobe-ohjelmistot ovat varmasti kaikille jollain tavalla tuttuja. Adobe on kehittänyt vuosien varrella loistavia ohjelmistoja niin kuvankäsittelyyn, videoeditointiin sekä äänimaailmaan. Adoben verkkosivut sisältävät ilmaisia opetusohjelmia ja videoita. (Adobe n.d-a.)

4.1.1 Käyttöliittymä ja sen käytettävyys

Adobe on mielestäni onnistunut erittäin hyvin käyttöliittymänsä suunnittelussa sisällyttämään yksinkertaisesti hyviä sekä helppoja tapoja muokata toimintoja. Ohjelman asetuksista käyttäjä voi halutessaan muokata graafisen käyttöliittymän värejä. Kaikki efektit, asetukset, raitojen lisääminen, näkymien piilottaminen ja esiintuominen, on helposti muokattavissa työkalupalkissa sijaitsevista painikkeista, joka sijaitsee ohjelman yläkulmassa.

Alla sijaitsevassa kuvassa on havainnollistettu Adobe Audition -ohjelmiston aloitusnäkymää (ks. Kuva 36), josta käyttäjä voi lähteä rakentamaan haluamaansa projektia. Aloitusnäkymässä näkyy projektin soittolista, johon käyttäjä voi liittää tai nauhoittaa tarvitsemiansa äänitiedostoja. Soittolistassa äänitiedostojen siirtely onnistuu vaivattomasti. Soittolistan alakulmassa sijaitsevat painikkeet toistolle, kelaukselle eteen- tai taaksepäin, pysäytykselle sekä nauhoitukselle. Aloitusnäkymässä näkyy myös efekti-ikkuna, joka sisältää valmistajan kehittämiä esiasetuksia erilaisille projekteille, jotka lisäävät tiettyjä efektejä projektiin.

Vasemmassa yläkulmassa ohjelma sisältää tiedostoikkunan, joka näyttää projektin käytössä olevat tiedostot. Vasemmassa alakulmassa sijaitsee projektin historia, joka näyttää käyttäjän jokaisen liikkeen projektissa. Projektin historiasta on helppo hallita toimintoja, kuten poistaa vanha toiminto, jonka käyttäjä toteaa tarpeettomaksi.



Kuva 36. Adobe Audition -ohjelmiston käyttöliittymä.

4.1.2 Yhteenveto

Adobe Audition -ohjelmiston rakenne musiikillisten projektien rakentamiseen ei ole niin laaja verrattuna FL Studio -ohjelmiston ominaisuuksiin sekä rakenteeseen. Äänitiedoston nauhoittaminen toki onnistuu, mutta ohjelma ei sisällä FL Studio -ohjelmiston tasoisia Piano Roll - tai muita vastaavia melodian rakentamiseen tarkoitettuja toimintoja. Ääniohjelmistona Audition on kuitenkin erittäin hyvä, sillä se on hieman helppokäyttöisempi ja yksinkertaisempi, joka sopii loistavasti tämän tapaisten ohjelmistojen parissa aloitteleville käyttäjille.

Ohjelmistossa loistava ominaisuus on Adoben Premiere -videoeditorin yhdistäminen keskenään. Premiere-videoeditointiohjelmistosta voidaan avata videon ääniraidan suoraan Audition-ohjelmistoon, editoida ääniraitaa käyttäjän haluamalla tavalla ja muutokset tallentuvat suoraan videon äänitiedostoon. Se helpottaa videoiden editoimisen työskentelyssä huomattavasti.

Audition-ohjelmiston valmistajien kehittämien efektien valikoima on laaja sekä toimiva. Audition-ohjelmisto sisältää valikoiman erilaisia kompressointi-, kaiku-, taajuuskorjain-, suodatin-, särö- sekä viive-efektejä. Efektien asettaminen eri kanaville tai yksittäiseen äänitiedostoon on yksinkertaista

Audition-ohjelmisto on äänitiedoston korjaukseen ja muokkaamiseen erinomainen työkalu. Audition-ohjelmiston äänieditorilla käyttäjän on erittäin helppo poistaa ylimääräisiä ääniä ja ohjelma on loistava työkalu videoeditoinnin kanssa yhdistettynä.

Alla sijaitsevassa taulukossa (ks. Taulukko 5) on havainnollistettu ohjelmistojen välisiä käyttöjärjestelmävaatimuksia. Kuten huomataan, Audition toimii ainoastaan 64-bittisellä järjestelmällä ja se vaatii enemmän RAM-muistia sekä kovalevytilaa toimiakseen. Mac-järjestelmille molemmat ohjelmistot ovat olemassa, mutta FL Studio -ohjelmiston versio on vasta beta-versio, joka on kehityksen alla eikä vielä virallisesti julkaistu. (Adobe n.d-b.)

Taulukko 5. Ohjelmistojen väliset käyttöjärjestelmävaatimukset.

	FL Studio	***Audition***
32-bit	***X***	
64-bit	***X***	***X***
Mac- järjestelmille	***X (Beta)***	***X***
RAM	***1 Gt***	***4 Gt***
Kovalevytila	***1 Gt***	***4 Gt***

4.2 **Ableton Live**

Ableton Live on FL Studio -ohjelmiston tapaan DAW-ohjelmisto. Live on suunniteltu nimensä veroisesti toimimaan parhaiten artistien live-esityksissä, mutta sen rakenne omaa hyvät ominaisuudet myös ääniprojektien rakentamiseen. Ableton Live -ohjelmistosta on saatavilla muutamia erilaisia versioita, joista työssä käsiteltävänä on Ableton Live 9 Lite. Erilaiset äänilaitteistojen tai MIDI-kontrollerien valmistajat jakavat kyseistä Ableton Live 9 Lite-versiota laitteidensa tukena ilmaiseksi. Kyseisen ohjelmiston lisenssi on hankittu äänikortin mukana (Focusrite 2i4 2nd Gen).

Ableton Live -ohjelmisto toimii hieman eri tavalla kuin muut vastaavat ohjelmistot. Sen rakenne vastaa hieman taulukkomaista mallia, johon käyttäjä lisää haluamansa äänitiedostot tai efektit. Ableton Live on erittäin mielenkiintoinen ja toimiva ohjelmisto. Ableton Live toimii myös erittäin hyvin MIDI-kontrollereiden kanssa. Ableton Live 9 Lite -versiossa on kuitenkin rajoituksensa verrattaessa muihin Ableton-ohjelmiston tuottamiin versioihin.

Ableton-ohjelmiston verkkosivustolta on saatavilla erilaisia äänitiedostopaketteja sekä instrumentteja ohjelmiston tueksi. Niitä on saatavilla niin ilmaiseksi kuin maksullisenakin.

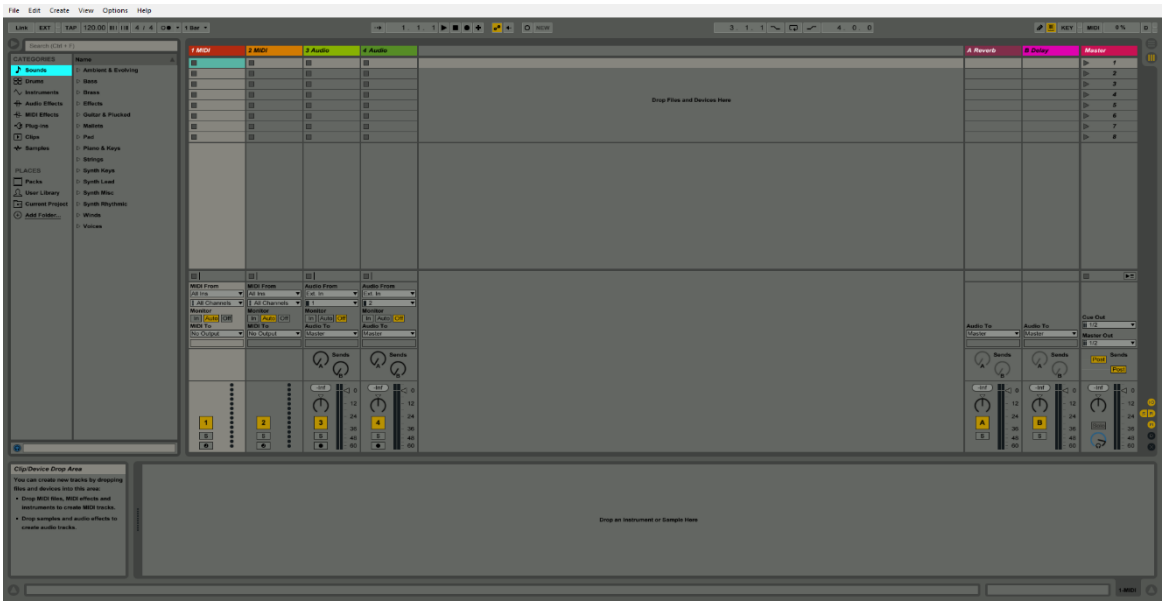
4.2.1 Käyttöliittymä ja sen käytettävyys

Käyttäjän avatessa Ableton Live 9 Lite -ohjelmiston avautuu näkymä, joka näkyy alla olevassa kuvassa (ks. Kuva 37). Yläreunassa sijaitsevista valikoista käyttäjä voi muokata asetuksia sekä projektia haluamakseen. Ikkunan vasemmassa reunassa sijaitsee selainikkuna. Käyttöliittymän keskellä on palkkeja, johon käyttäjä voi lisätä efektejä, instrumentteja sekä äänitiedostoja.

Palkkien alapuolella sijaitsee valikoita ja säätimiä, joista käyttäjä voi muokata äänen sisään- ja ulostuloja sekä raitojen äänenvoimakkuuksia. Palkkien alapuolella sijaitsee myös säätimet, joista käyttäjä voi lähettää äänen toiselle kanavalle. Käyttöliittymän alakulmassa käyttäjä voi halutessaan muokata lisäämiään efektejä. Äänitiedoston muokkaaminen onnistuu myös samaisessa laatikossa.

Graafiselta osuudeltaan käyttöliittymä on toteutettu yksinkertaisesti FL Studio -ohjelmistoon verrattaessa. Vertailukohteista tämä muistuttaa eniten Adobe-ohjelmistojen graafista käyttöliittymää. Värimaailmaa käyttäjä voi muuttaa mieleisekseen asetuksien kautta. Siirtymät eri ikkunoihin onnistuvat joko pikanäppäimillä näppäimistön kautta tai hiiren painikkeella kyseisen kuvakkeen kohdalta.

Käyttäjän rakentaessa projektiaan ohjelman kaikki ominaisuudet ovat helposti käytettävissä ilman ylimääräistä säätöä. Käyttäjä voi halutessaan piilottaa haluamansa kohteet yhdellä hiiren painalluksella ja tuoda ne samalla tavalla takaisin. (Ableton n.d-a.)



Kuva 37. Ableton Live -ohjelmiston käyttöliittymä.

4.2.2 Yhteenveto

FL Studio ja Ableton Live ovat hyvin samankaltaisia ohjelmistoja. Molemmat ohjelmistot ovat loistavia melodian rakentamisen suhteen, sillä molemmat sisältävät Piano Roll -työkalun. MIDI-kontrollerien käyttö on myös erittäin sulavaa ja helppoa. Ableton-ohjelmistoa olen käyttänyt huomattavasti lyhyemmän aikaa kuin FL Studio -ohjelmistoa, mutta sain ohjelmistosta peruselementit ja sen käytön hyvinkin nopeasti haltuun, joten puhumme siis hyvin yksinkertaisesta ja helposti haltuun otettavasta ohjelmistosta.

Kyseiseen vertailuun ei saatu Ableton Live -ohjelmistosta kaikkea mahdollista irti, koska käytössä on vain Ableton Live 9 Lite -versio. Ohjelmistoissa äänen laatu ja työnjälki ovat lähtökohtaisesti samanlaisia sekä yhtä hyviä, mutta hyvin pieniä eroja on havaittavissa ääniaaltoja analysoitaessa.

Ableton Live -ohjelmiston rakentamisessa on selkeästi pyritty panostamaan live-soiton puolelle. Erilaisten live-projektien rakentaminen on mielestäni helpompaa kuin vastaavanlaisen projektin rakentaminen FL Studio -ohjelmiston puolella. Ableton Live sopii myös toki muun tapaisten projektien rakentamiseen.

Alla sijaitsevassa taulukossa (ks. Taulukko 6) on havainnollistettu ohjelmistojen välisiä käyttöjärjestelmien eroja. Siitä voidaan havaita, että Ableton Live vaatii hieman enemmän RAM-muistia ja kovalevytilaa FL Studio -ohjelmistoon verrattuna. Mac-järjestelmille molemmat ohjelmistot ovat olemassa, mutta FL Studio -ohjelmiston versio on vasta beta-versio, joka on kehityksen alla eikä vielä virallisesti julkaistu. (Ableton n.d-b.)

Taulukko 6. Ohjelmistojen väliset käyttöjärjestelmävaatimukset.

	FL Studio	*Ableton Live 9 Lite*
32-bit	***X***	***X***
64-bit	***X***	***X***
Mac- järjestelmille	***X (Beta)***	***X***
RAM	***1 Gt***	***4 Gt***
Kovalevytila	***1 Gt***	***3 Gt***

4.3 **Pro Tools**

Pro Tools on Avid-yhtiön kehittämä ääniohjelmisto ja sen ensimmäiset versiot saapuivat markkinoille vuonna 2010. Oma käyttökokemukseni Pro Tools -ohjelmistosta ei ole aikaisemmin mainitsemani Ableton Live -ohjelmiston tapaan millään tavalla pitkä, mutta olen opiskellut ohjelmistoa juuri tätä kyseistä työtä varten ja olen todennut sen erittäin yksinkertaiseksi sekä helposti opittavaksi ohjelmistoksi.

Pro Tools sisältää samankaltaisen Piano Roll -työkalun melodian rakentamiseen FL Studio - ja Ableton Live -ohjelmistojen tapaan, mikä on mielestäni erittäin positiivinen ominaisuus jokaisessa ääniohjelmistossa. Efektien ja instrumenttien lisääminen raidalle on tehty yksinkertaiseksi ja helpoksi.

Pro Tools on myöskin lisenssipohjainen ohjelmisto, ja Avid tarjoaa kyseistä First-versiota ilmaiseksi. Työssä käytetään Pro Tools First -versiota. Kyseiseen ohjelmistoon saa lisenssin rekisteröitymällä Avid-yhtiön asiakkaaksi sen verkkosivustolla. Pro Tools First on kaikista rajoitetuin versio, mikä ilmenee instrumenttien puutteellisuudessa sekä projektien lukumäärässä.

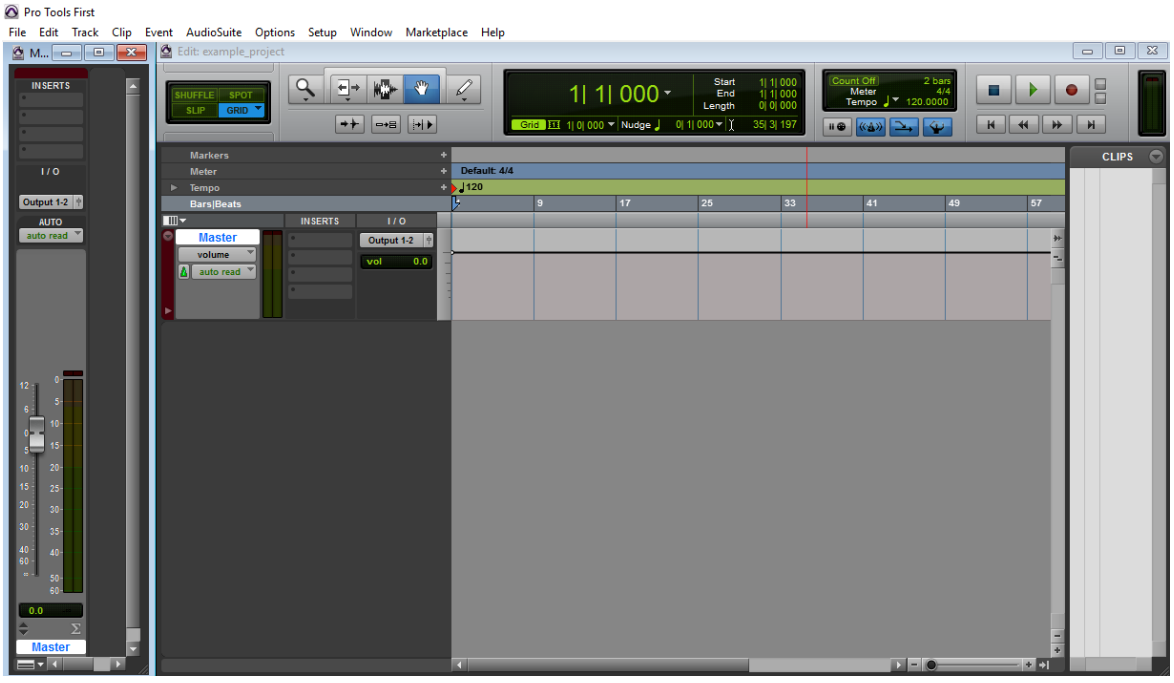
4.3.1 Käyttöliittymä ja sen käytettävyys

Aloittaessaan ohjelman käytön käyttäjän on luotava uusi projekti. Projektin voi luoda joko täysin tyhjään pohjaan tai erilaisiin valmiisiin esiasetuksiin, jotka ovat suunniteltuja erilaisiin projekteihin. First-versiossa on mahdollista olla kolme projektia samanaikaisesti luotuna, muissa versiossa

tämä ominaisuus on rajoittamaton. Projektien rajan tullessa täyteen käyttäjän on poistettava jokin luomistaan projekteista ja tehtävä tilaa uudelle projektille.

Alla sijaitsevassa kuvassa nähdään ohjelmiston käyttöliittymä (ks. Kuva 38). Kuvan yläreunassa sijaitsee työkalupalkki, josta käyttäjä voi hallinnoida projektiaan. Kuvan vasemmassa reunassa sijaitsee mikseri, josta käyttäjä voi hallinnoida projektinsa äänitiedostojen äänenvoimakkuuksia. Mikserin oikealla puolella taas sijaitsee editointiin tarkoitettu ikkuna. Käyttäjä voi editoida äänitiedostoa, asettaa uusia kanavia, säätää kanavien sisään- ja ulostuloja sekä asettaa instrumentteja ja efektejä projektilleen.

Editointi-ikkunan oikealla puolella on palkki, johon muodostuu kaikki käyttäjän käyttämät tiedostot. Editointi-ikkunan yläpuolella sijaitsee aikajana, jonka voi asettaa näyttämään ajan joko palkkeina tai sitten minuutteina ja sekunteina. Aikajanasta hallinnoidaan myös projektin tempoa. Aikajanan vasemmalta puolelta löytyy painikkeita, joilla käyttäjä voi editoida äänitiedostoa halutessaan erittäin tarkasti ja laajemmalla asteella. Aikajanan oikealta puolelta löytyy painikkeet pysäytykselle, toistolle, äänittämiselle sekä kelaukselle eteen- tai taaksepäin. (Avid n.d-a.)



Kuva 38. Pro Tools -ohjelmiston käyttöliittymä.

4.3.2 Yhteenveto

FL Studio -ohjelmistoon verrattaessa Pro Tools ei ole yhtä monipuolinen ja kattava ohjelmisto. Tässä vaiheessa on toki huomioitava, että kyseinen Pro Tools First -versio on ilmainen. Ilmaiseksi versioksi Pro Tools First on erittäin hyvä, ja Pro Tools -ohjelmiston normaaliversio toisi lisää kattavuutta

työhön verrattaessa Pro Tools First -ilmaisversioon. Pro Tools First -versio antaa aloittelevalle ääniohjelmistojen käyttäjälle hyvän pohjan harjoitella ääniprojektien luomista sekä musiikin tekoa.

Pro Tools -ohjelmistot saavat ison miinuksen VST-liitännäisten puuttumisen takia. Varsinkin Pro Tools First-versiossa, koska se ei tue kaikkia AAX Plugin -liitännäisiä, jotka toimisivat Pro Tools -normaaliversiossa. Miinusta saa myöskin projektien määrä, joka on rajoitettu vain kolmeen ja äänitiedoston luomisessa käytettävät äänitiedostotyypit, joita oli valittavissa vain kaksi.

Pro Tools -ohjelmisto sisältää hyvät työkalut äänitiedoston editointiin, melodian rakentamiseen (Piano Roll -työkalu) sekä äänittämiseen. Käyttöliittymä on rakennettu yksinkertaiseksi ja perusasiat löytyvät helposti.

Alla sijaitsevassa taulukossa (ks. Taulukko 7) on havainnollistettu ohjelmistojen välisiä käyttöjärjestelmien eroja. Siitä voidaan havaita, että Pro Tools First -ohjelmisto vaatii hieman vähemmän RAM-muistia ja todella paljon enemmän kovalevytilaa FL Studio -ohjelmistoon verrattuna. Pro Tools -ohjelmistolla ei ole tukea 32-bittiselle käyttöjärjestelmälle. Mac-järjestelmille molemmat ohjelmistot ovat olemassa, mutta FL Studio -ohjelmiston versio on vasta beta-versio, joka on kehityksen alla eikä vielä virallisesti julkaistu. (Avid n.d-b.)

Taulukko 7. Ohjelmistojen väliset käyttöjärjestelmävaatimukset.

	FL Studio	***Pro Tools First***
32-bit	***X***	
64-bit	***X***	***X***
Mac- järjestelmille	***X (Beta)***	***X***
RAM	***1 Gt***	***2 Gt***
Kovalevytila	***1 Gt***	***15 Gt***

5 OHJELMISTOJEN VÄLINEN ÄÄNINÄYTTEIDEN ANALYSOINTI

Tässä vaiheessa työtä yritetään tutkia ja havainnollistaa, löytyykö tietyn äänitiedoston rakenteessa eroja eri ohjelmistojen välillä analysoimalla äänitiedostoa eri taajuusasteikoilla sekä tutkia, sisältävätkö ne eroja äänenvoimakkuuden suhteen siihen tarkoitetulla ohjelmalla. Lähtökohta on se, että saman äänitiedoston rakenne eri ohjelmistojen välillä on ainakin todella lähellä samaa, eikä ihmiskorva sitä havaitse. Jos eroja esiintyy, ne ovat todella pieniä ja huomattavissa vain ääninäytteen analysointiin tarkoitetuilla työkaluilla.

Ääninäytteen analysoimisessa käytetään neljää erilaista ääninäytettä, jotka ovat FL Studio -ohjelmiston aloitusnäkymään esiasetettuja rumpuääniä. Ääninäytteet ovat ”Kick”-, ”Clap”-, ”Hat”- ja ”Snare”-rumpuäänet, joista on luotu uusi ja yksinkertainen äänitiedosto jokaisen ohjelman omalla äänitiedoston luomiseen tarkoitetulla työkalulla.

Lähes jokaisessa tutkimuksessa on tapana esiintyä pieniä ongelmia, niin myös tässä. Käyttämäni työkalu ääninäytteen analysointiin olisi sisältänyt AAX Plugin -liitännäisen, mutta Pro Tools First -versio ei tätä kyseistä liitännäistä tue lainkaan. Jouduin siis jättämään Pro Tools -ohjelmiston kokonaan ulkopuolelle kyseisestä tutkimuksesta.

5.1 Flux Studio Session Analyzer

Flux Studio Session Analyzer -ohjelma (ks. Kuva 39) on ääninäytteen analysointiin tarkoitettu työkalu. Kyseessä on ohjelma, jolla käyttäjä voi seurata äänitiedoston kulkua reaaliajassa erilaisilla mittareilla. Flux Studio Session Analyzer on maksullinen ohjelma, johon on hankittu lisenssi äänikortin mukana. Ohjelma toimii siten, että käyttäjä asettaa ääniohjelmistonsa yhdelle kanavalle SampleGrabber-liitännäisen, joka ohjaa äänen sitä käyttäen ääniohjelmistosta suoraan Studio Session Analyzer -ohjelmaan.

Alla sijaitsevassa kuvassa on havainnollistettu Studio Session Analyzer -ohjelman toiminta. Ohjelma sisältää muutaman erilaisen mittarin, joista käytän ainoastaan ”Magnitude Spectrum”-mittaria, joka sijaitsee ikkunan oikeassa yläkulmassa. Se näyttää äänitiedoston rakenteen ja desibelien korkeuksien arvot eri taajuusasteikoilla, jotka ovat 20Hz - 20 000Hz.

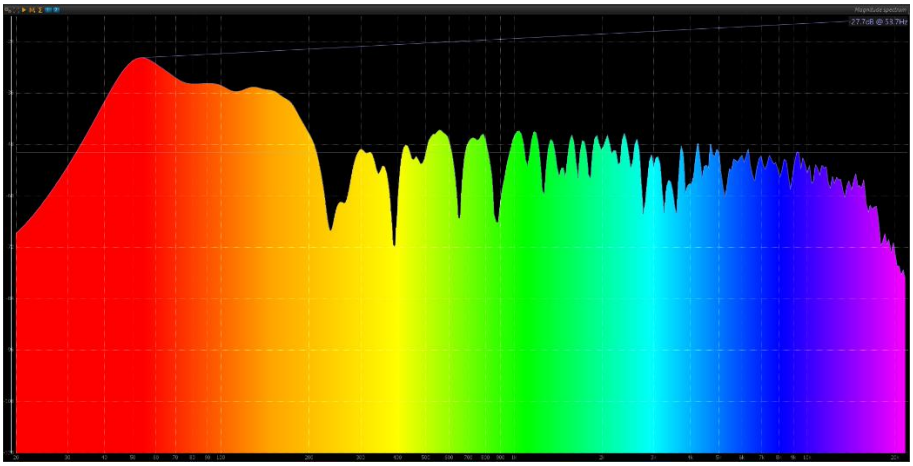
Kuva 39. Studio Session Analyzer -ohjelma.

5.2 **Magnitude Spectrum -mittari**

Tutkiessa ääninäytteitä käytetään Studio Session Analyzer -ohjelman Magnitude Spectrum -mittaria ja alla sijaitsevassa kuvassa on havainnollistettu sen toiminta, ja mittariin on asetettu ”Kick Drum”-ääninäyte (ks. Kuva 40). Mittari näyttää sen vasemmassa reunassa desibelien voimakkuuksien arvot, jotka mittari näyttää pystysuoraan. Mittarin alareunassa sijaitsevat taajuusasteikot 20Hz - 20 000Hz, jotka muodostuvat mittarissa vaakasuoraan. Näistä muodostuu alla sijaitsevassa kuvassa näkyvä ääniaalto.

Mittari näyttää erikseen myös kaikista korkeimman desibelin arvon erikseen, sekä taajuusasteikon jolla arvo sijaitsee. Käyttäjällä on mahdollisuus muokata asetuksia mittarin sisällä.

Ääninäytteiden erot tulevat olemaan hyvin pieniä, ja alla sijaitseva esimerkki ei näytä eroja juuri lainkaan. Ääninäytteitä on tutkittu mahdollisimman tarkasti ja tarkennettu kaikista tarkimpaan mahdolliseen desibelin arvoon, jota mittari voi jokaisen taajuusasteikon kohdalla näyttää. Tutkimuksen tulokset on havainnollistettu taulukoina sekä kaavioina, joista erot ovat huomattavissa selkeämmin jokaisen ääninäytteen kohdalla.



Kuva 40. Magnitude Spectrum -mittari sisältäen ”Kick Drum”-ääninäytteen.

5.3 **Kick Drum -ääninäyte**

Taulukoista voidaan havaita (ks. Taulukko 8 ja Taulukko 9), että ”Kick Drum”-ääninäytteissä on eroja ohjelmistojen välillä. Etenkin Adobe Audition eroaa kahdesta muusta ohjelmistosta jokaisella taajuusalueella, kun taas FL Studio sekä Ableton Live ovat erittäin lähellä toistensa arvoja. Suurimmat eroavuudet ”Kick Drum”-ääninäytteestä saaduissa arvoissa ohjelmistojen välillä ovat taajuusalueilla 400 Hz, 900 Hz sekä 9 000 Hz.

Taulukko 8. ”Kick Drum”-ääninäytteen desibelien korkeusarvot eri hertsiasteikoilla taulukkomuodossa.

	FL Studio	*Ableton*	*Audition*
20 Hz	**-68,2 dB**	**-68,7 dB**	**-71,8 dB**
30 Hz	**-53,1 dB**	**-53,1 dB**	**-56,1 dB**
40 Hz	**-38,1 dB**	**-38,1 dB**	**-41,1 dB**
50 Hz	**-28,4 dB**	**-28,4 dB**	**-31,5 dB**
60 Hz	**-29,1 dB**	**-29,1 dB**	**-32,1 dB**
70 Hz	**-32,3 dB**	**-32,4 dB**	**-35,3 dB**
80 Hz	**-33,8 dB**	**-33,8 dB**	**-36,8 dB**
90 Hz	**-33,7 dB**	**-33,7 dB**	**-36,7 dB**
100 Hz	**-34,1 dB**	**-34,1 dB**	**-37,2 dB**
200 Hz	**-45,3 dB**	**-45,3 dB**	**-48,4 dB**
300 Hz	**-49,1 dB**	**-48,9 dB**	**-52,1 dB**
400 Hz	**-64,5 dB**	**-57,2 dB**	**-67,6 dB**
500 Hz	**-50,2 dB**	**-48,6 dB**	**-53,2 dB**
600 Hz	**-46,8 dB**	**-48,0 dB**	**-49,9 dB**
700 Hz	**-47,4 dB**	**-48,1 dB**	**-50,5 dB**
800 Hz	**-46,7 dB**	**-47,4 dB**	**-49,7 dB**
900 Hz	**-61,6 dB**	**-57,1 dB**	**-64,6 dB**
1 000 Hz	**-46,3 dB**	**-46,6 dB**	**-49,3 dB**
2 000 Hz	**-49,0 dB**	**-52,5 dB**	**-52,0 dB**
3 000 Hz	**-52,7 dB**	**-56,7 dB**	**-55,7 dB**
4 000 Hz	**-57,1 dB**	**-59,2 dB**	**-60,1 dB**
5 000 Hz	**-49,6 dB**	**-53,5 dB**	**-52,6 dB**
6 000 Hz	**-50,5 dB**	**-51,3 dB**	**-53,5 dB**
7 000 Hz	**-50,9 dB**	**-51,3 dB**	**-53,9 dB**
8 000 Hz	**-54,3 dB**	**-56,5 dB**	**-57,3 dB**
9 000 Hz	**-53,9 dB**	**-61,3 dB**	**-56,9 dB**
10 000 Hz	**-54,4 dB**	**-55,2 dB**	**-57,4 dB**
20 000 Hz	**-73,1 dB**	**-75,1 dB**	**-76,2 dB**

Taulukko 9. ”Kick Drum”-ääninäytteen desibelien korkeusarvot eri hertsiasteikoilla kaaviomuodossa.

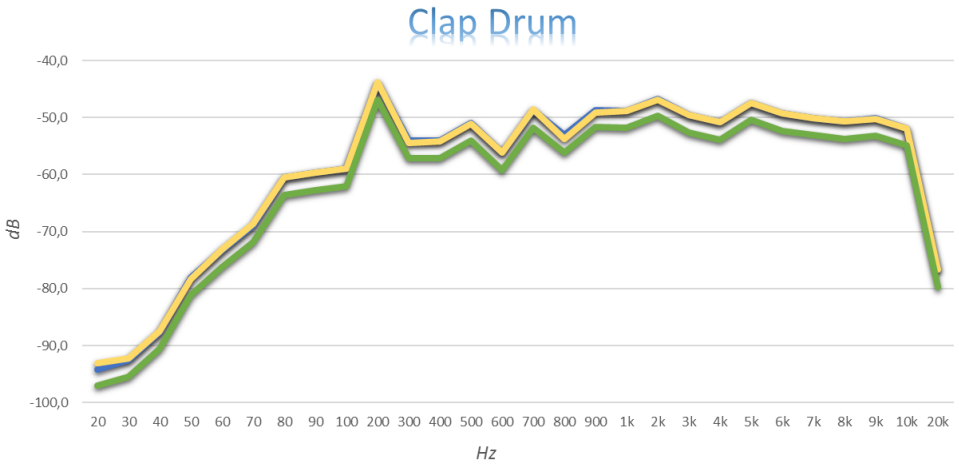
5.4 **Clap Drum -ääninäyte**

Taulukoista voidaan havaita (ks. Taulukko 10 ja Taulukko 11), että ”Clap Drum”-ääninäytteissä on eroja ohjelmistojen välillä. Etenkin Adobe Audition eroaa kahdesta muusta ohjelmistosta jokaisella taajuusalueella, kun taas FL Studio sekä Ableton Live ovat erittäin lähellä toistensa arvoja. ”Clap Drum”-ääninäytteessä ei ole suuria eroavuuksia ohjelmistojen välillä tietyillä taajuusalueilla.

Taulukko 10. ”Clap Drum”-ääninäytteen desibelien korkeusarvot eri hertsiasteikoilla taulukkomuodossa.

	FL Studio	*Ableton*	*Audition*
20 Hz	**-94,2 dB**	**-93,1 dB**	**-97,1 dB**
30 Hz	**-92,5 dB**	**-92,3 dB**	**-95,5 dB**
40 Hz	**-87,6 dB**	**-87,5 dB**	**-90,6 dB**
50 Hz	**-78,1 dB**	**-78,4 dB**	**-81,1 dB**
60 Hz	**-73,3 dB**	**-73,2 dB**	**-76,3 dB**
70 Hz	**-69,0 dB**	**-68,7 dB**	**-71,9 dB**
80 Hz	**-60,5 dB**	**-60,5 dB**	**-63,6 dB**
90 Hz	**-59,7 dB**	**-59,7 dB**	**-62,7 dB**
100 Hz	**-59,0 dB**	**-59,0 dB**	**-62,1 dB**
200 Hz	**-43,9 dB**	**-43,8 dB**	**-46,9 dB**
300 Hz	**-54,1 dB**	**-54,4 dB**	**-57,2 dB**
400 Hz	**-54,1 dB**	**-54,2 dB**	**-57,2 dB**
500 Hz	**-51,0 dB**	**-51,1 dB**	**-54,0 dB**
600 Hz	**-56,2 dB**	**-56,2 dB**	**-59,2 dB**
700 Hz	**-48,7 dB**	**-48,5 dB**	**-51,8 dB**
800 Hz	**-53,1 dB**	**-53,8 dB**	**-56,2 dB**
900 Hz	**-48,7 dB**	**-49,1 dB**	**-51,6 dB**
1 000 Hz	**-48,8 dB**	**-48,9 dB**	**-51,8 dB**
2 000 Hz	**-46,7 dB**	**-46,9 dB**	**-49,7 dB**
3 000 Hz	**-49,6 dB**	**-49,6 dB**	**-52,6 dB**
4 000 Hz	**-50,8 dB**	**-50,8 dB**	**-53,9 dB**
5 000 Hz	**-47,4 dB**	**-47,5 dB**	**-50,4 dB**
6 000 Hz	**-49,3 dB**	**-49,3 dB**	**-52,3 dB**
7 000 Hz	**-50,1 dB**	**-50,1 dB**	**-53,1 dB**
8 000 Hz	**-50,7 dB**	**-50,7 dB**	**-53,7 dB**
9 000 Hz	**-50,1 dB**	**-50,2 dB**	**-53,2 dB**
10 000 Hz	**-51,9 dB**	**-51,9 dB**	**-54,9 dB**
20 000 Hz	**-76,6 dB**	**-76,7 dB**	**-79,7 dB**

Taulukko 11. ”Clap Drum”-ääninäytteen desibelien korkeusarvot eri hertsiasteikoilla kaaviomuodossa.



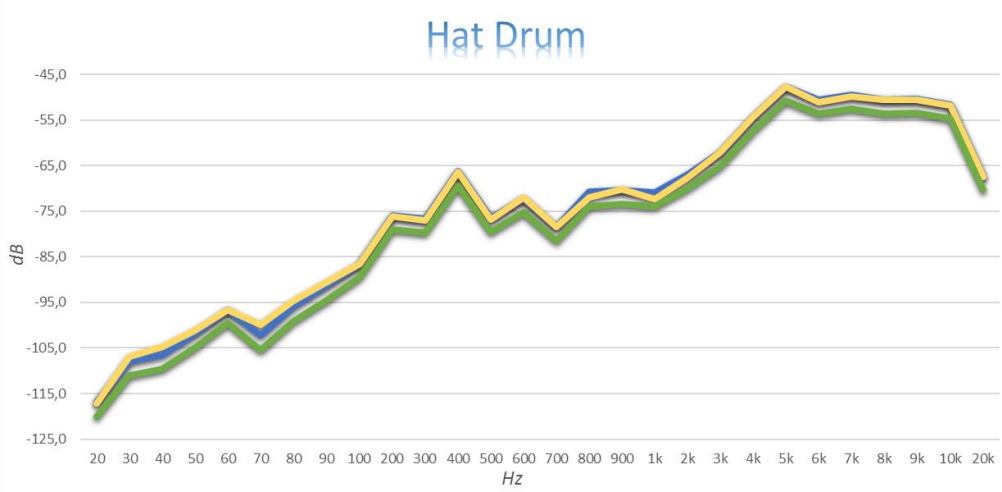
5.5 Hat Drum -ääninäyte

Taulukoista voidaan havaita (ks. Taulukko 12 ja Taulukko 13), että ”Hat Drum”-ääninäytteissä on eroja ohjelmistojen välillä. Etenkin Adobe Audition eroaa kahdesta muusta ohjelmistosta jokaisella taajuusalueella, kun taas FL Studio sekä Ableton Live ovat erittäin lähellä toistensa arvoja. ”Hat Drum”-ääninäytteessä ei ole suuria eroavuuksia ohjelmistojen välillä tietyillä taajuusalueilla ja suurimmat eroavuudet ohjelmistojen välillä ovat taajuusalueilla 40 Hz sekä 70 Hz.

Taulukko 12. ”Hat Drum”-ääninäytteen desibelien korkeusarvot eri hertsiasteikoilla taulukkomuodossa.

	FL Studio	*Ableton*	*Audition*
20 Hz	**-116,8 dB**	**-117,1 dB**	**-120,0 dB**
30 Hz	**-108,0 dB**	**-106,9 dB**	**-111,1 dB**
40 Hz	**-106,5 dB**	**-104,7 dB**	**-109,6 dB**
50 Hz	**-101,8 dB**	**-101,1 dB**	**-104,9 dB**
60 Hz	**-96,6 dB**	**-96,6 dB**	**-99,6 dB**
70 Hz	**-102,3 dB**	**-99,9 dB**	**-105,4 dB**
80 Hz	**-96,0 dB**	**-94,6 dB**	**-99,1 dB**
90 Hz	**-91,2 dB**	**-90,6 dB**	**-94,4 dB**
100 Hz	**-86,5 dB**	**-86,6 dB**	**-89,5 dB**
200 Hz	**-76,0 dB**	**-76,2 dB**	**-79,0 dB**
300 Hz	**-76,6 dB**	**-77,0 dB**	**-79,8 dB**
400 Hz	**-66,3 dB**	**-66,4 dB**	**-69,4 dB**
500 Hz	**-76,4 dB**	**-76,8 dB**	**-79,5 dB**
600 Hz	**-72,2 dB**	**-71,9 dB**	**-75,3 dB**
700 Hz	**-78,5 dB**	**-78,4 dB**	**-81,5 dB**
800 Hz	**-70,8 dB**	**-71,9 dB**	**-73,9 dB**
900 Hz	**-70,4 dB**	**-70,1 dB**	**-73,5 dB**
1 000 Hz	**-70,9 dB**	**-72,4 dB**	**-74,0 dB**
2 000 Hz	**-66,9 dB**	**-67,6 dB**	**-70,0 dB**
3 000 Hz	**-62,1 dB**	**-62,1 dB**	**-65,2 dB**
4 000 Hz	**-54,4 dB**	**-54,4 dB**	**-57,4 dB**
5 000 Hz	**-47,6 dB**	**-47,6 dB**	**-50,7 dB**
6 000 Hz	**-50,5 dB**	**-51,1 dB**	**-53,6 dB**
7 000 Hz	**-49,4 dB**	**-49,9 dB**	**-52,5 dB**
8 000 Hz	**-50,6 dB**	**-50,6 dB**	**-53,7 dB**
9 000 Hz	**-50,4 dB**	**-50,6 dB**	**-53,5 dB**
10 000 Hz	**-51,7 dB**	**-51,8 dB**	**-54,7 dB**
20 000 Hz	**-67,4 dB**	**-67,5 dB**	**-70,4 dB**

Taulukko 13. ”Hat Drum”-ääninäytteen desibelien korkeusarvot eri hertsiasteikoilla kaaviomuodossa.



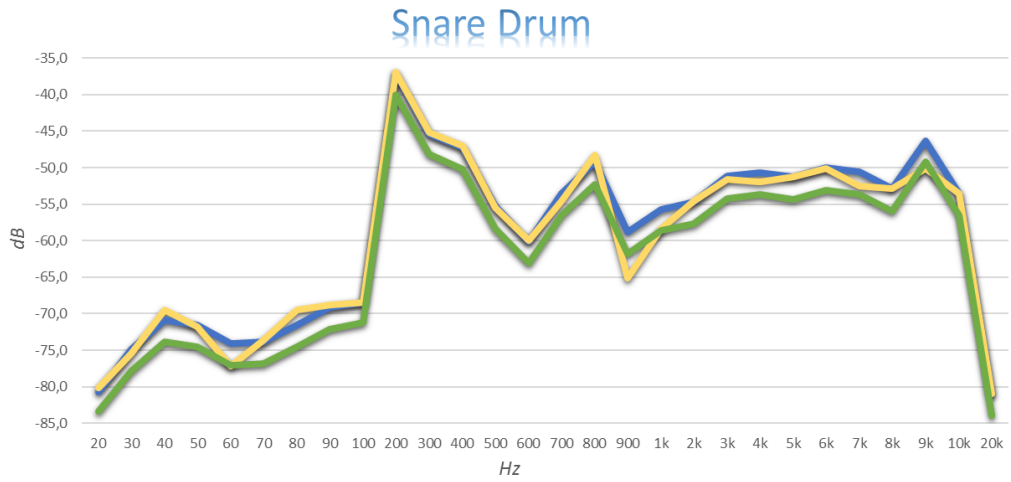
5.6 **Snare Drum -ääninäyte**

Taulukoista voidaan havaita (ks. Taulukko 14 ja Taulukko 15), että ”Snare Drum”-ääninäytteissä on eroja ohjelmistojen välillä. Etenkin Adobe Audition eroaa kahdesta muusta ohjelmistosta jokaisella taajuusalueella, kun taas FL Studio sekä Ableton Live ovat erittäin lähellä toistensa arvoja. Suurimmat eroavuudet ”Snare Drum”-ääninäytteestä saaduissa arvoissa ohjelmistojen välillä ovat taajuusalueilla 80 Hz, 900 Hz sekä 9 000 Hz.

Taulukko 14. ”Snare Drum”-ääninäytteen desibelien korkeusarvot eri hertsiasteikoilla taulukkomuodossa.

	FL Studio	*Ableton*	*Audition*
20 Hz	**-80,7 dB**	**-80,2 dB**	**-83,4 dB**
30 Hz	**-74,9 dB**	**-75,5 dB**	**-77,9 dB**
40 Hz	**-70,9 dB**	**-69,5 dB**	**-73,9 dB**
50 Hz	**-71,5 dB**	**-71,8 dB**	**-74,5 dB**
60 Hz	**-74,1 dB**	**-77,2 dB**	**-77,0 dB**
70 Hz	**-73,9 dB**	**-73,6 dB**	**-76,8 dB**
80 Hz	**-71,6 dB**	**-69,5 dB**	**-74,5 dB**
90 Hz	**-69,1 dB**	**-68,8 dB**	**-72,1 dB**
100 Hz	**-68,4 dB**	**-68,4 dB**	**-71,2 dB**
200 Hz	**-37,0 dB**	**-36,9 dB**	**-40,0 dB**
300 Hz	**-45,4 dB**	**-45,2 dB**	**-48,2 dB**
400 Hz	**-47,2 dB**	**-47,0 dB**	**-50,2 dB**
500 Hz	**-55,3 dB**	**-55,5 dB**	**-58,4 dB**
600 Hz	**-60,1 dB**	**-60,0 dB**	**-63,1 dB**
700 Hz	**-53,6 dB**	**-54,6 dB**	**-56,5 dB**
800 Hz	**-49,2 dB**	**-48,3 dB**	**-52,3 dB**
900 Hz	**-58,8 dB**	**-65,1 dB**	**-61,8 dB**
1 000 Hz	**-55,7 dB**	**-58,4 dB**	**-58,6 dB**
2 000 Hz	**-54,7 dB**	**-54,6 dB**	**-57,7 dB**
3 000 Hz	**-51,1 dB**	**-51,6 dB**	**-54,2 dB**
4 000 Hz	**-50,7 dB**	**-51,9 dB**	**-53,7 dB**
5 000 Hz	**-51,3 dB**	**-51,3 dB**	**-54,3 dB**
6 000 Hz	**-50,0 dB**	**-50,1 dB**	**-53,1 dB**
7 000 Hz	**-50,6 dB**	**-52,5 dB**	**-53,7 dB**
8 000 Hz	**-52,9 dB**	**-52,9 dB**	**-55,9 dB**
9 000 Hz	**-46,3 dB**	**-50,1 dB**	**-49,2 dB**
10 000 Hz	**-53,5 dB**	**-53,5 dB**	**-56,5 dB**
20 000 Hz	**-80,9 dB**	**-80,9 dB**	**-83,9 dB**

Taulukko 15. ”Snare Drum”-ääninäytteen desibelien korkeusarvot eri hertsiasteikoilla kaaviomuodossa.



5.7 Ääninäytteiden analysoinnin yhteenveto

Kuten äskettäin mainittiin, jokainen ohjelmisto antoi tutkimuksessa erilaisia arvoja ääninäytteille. Toiset jyrkemmin, toiset tasaisemmin. Adobe Audition antoi jokaisessa ääninäytteessä erilaisen tuloksen, kuin FL Studio tai Ableton Live ja eroavuuden määrä oli lähtökohtaisesti noin kolmen desibelin verran jokaisella taajuusasteikolla.

FL Studio - ja Ableton Live -ohjelmistojen arvot olivat lähes identtiset tai ainakin erittäin lähellä toisiaan jokaisessa ääninäytteessä. Ääninäytteiden välillä tuloksien samankaltaisuus paistoi esille ”Hat Drum”- ja ”Clap Drum”-näytteissä, kun taas ”Kick Drum”- ja ”Snare Drum”-näytteiden arvot olivat hieman enemmän eroavaisia kaavioihin rakentuneiden tulosten perusteella. Eroavuuksia syntyi eniten taajuusalueilla 900 Hz ja 9000 Hz.

6 YHTEENVETO

Työssä pyrittiin tutkimaan ääniohjelmistojen eroavaisuuksia. Tutkimustulokset antoivat ilmi, että eri ohjelmistojen välillä tehdyt tutkimukset antoivat pieniä eroavaisuuksia äänitiedostojen rakenteessa. Tutkimustuloksista saatujen arvojen eroavaisuudet olivat muutamien desibelien luokkaa, mutta siltikään näiden äänitiedostojen eroavaisuuksia ei huomata ihmiskorvalla vaan ainoastaan tutkimalla niiden arvoja erittäin tarkasti.

Eri ohjelmistojen vertailu keskenään osoittautui odotusten mukaiseksi, sillä pääpiirteittäin ohjelmistot ovat hyvin samanlaisia keskenään. Eroavaisuuksia löytyi hieman käyttöjärjestelmien vaatimuksista.

Tutkimus oli erittäin mielenkiintoinen ja opettavainen, sillä työhön saatiin havainnollistettua tutkimustuloksia, joita olivat epäselviä ennen työn aloittamista. Tulevaisuudessa tutkimustuloksia voidaan hyödyntää erilaisten projektien parissa. Allekirjoittaneelle projekti oli erittäin hyvää opetusta tulevaisuutta varten, sillä tarkoitus on syventää osaamista samantapaisten projektien parissa.

LÄHTEET

Ableton (n.d-a.) Ableton Live. Viitattu 09.03.2017
https://www.ableton.com/en/live/

Ableton (n.d-b.) Ableton Live System Requirements. Viitattu 14.03.2017
https://help.ableton.com/hc/en-us/articles/209773025-Live-9-Minimum-System-Requirements

Adobe (n.d-a.) Adobe Audition CC. Viitattu 02.02.2017
https://www.adobe.com/fi/products/audition.html

Adobe (n.d-b.) Adobe Audition System Requirements. Viitattu 04.02.2017
https://helpx.adobe.com/audition/system-requirements.html

Avid (n.d-a.) Pro Tools First. Viitattu 02.04.2017
http://www.avid.com/pro-tools-first

Avid (n.d-b.) Pro Tools System Requirements. Viitattu 15.04.2017
http://www.avid.com/pro-tools/compare#Pro-Tools-Minimum-System-Requirements

Image-Line (n.d-a.) FL Studio 12. Viitattu 15.12.2016.
https://www.image-line.com/flstudio/

Image-Line (n.d-b.) A Company History. Viitattu 15.12.2016
http://www.image-line.com/company/

Image-Line (n.d-c.) FL Studio 12 Reference Manual. Viitattu 20.12.2016
https://www.image-line.com/support/FLHelp/

Image-Line (n.d-d.) FL Studio System Requirements. Viitattu 27.12.2016
https://www.image-line.com/downloads/flstudiodownload.html

JUCE (n.d-a.) Projucer. Viitattu 28.11.2017
https://juce.com/projucer

JUCE (n.d-b.) Made with JUCE. Viitattu 28.11.2017
https://juce.com/made-with-juce

JUCE (n.d-c.) Tutorials. Viitattu 28.11.2017
https://juce.com/tutorials

JUCE (n.d-d.) Get JUCE. Viitattu 28.11.2017
https://juce.com/get-juce

Steinberg (n.d-a.) Developers. Viitattu 28.11.2017
https://www.steinberg.net/en/company/developers.html

Liite 1

PLUGINEDITOR.CPP

```
#include "PluginProcessor.h"
#include "PluginEditor.h"

AudioGainAudioProcessorEditor::AudioGainAudioProcessorEditor
(AudioGainAudioProcessor& p)
                : AudioProcessorEditor(&p), processor(p)

{
//Määritetään ikkunan koko
setSize(400, 300);

//Määritetään yhteys prosessorin ja säätimen välille
sliderAttach = new AudioProcessorValueTreeState::SliderAttachment
(processor.parameters, GAIN_ID, gainSlider);

//Määritetään säätimen tyyli
gainSlider.setSliderStyle(Slider::SliderStyle::RotaryHorizontalVerticalDrag);

//Määritetään teksti- ikkunan koko ja tyyli
gainSlider.setTextBoxStyle(Slider::TextBoxBelow, true, 100, 30);

//Määritetään arvon perään teksti ”dB”
gainSlider.setTextValueSuffix(" dB");

//Määritetään arvoalue (minimi, maksimi)
gainSlider.setRange(-48.0f, 0.0f);

//Määritetään kuuntelija toimimaan säätimen arvojen muuttuessa
gainSlider.addListener(this);

//Määritetään säädin näkyviin ikkunaan
addAndMakeVisible(gainSlider);
}


AudioGainAudioProcessorEditor::~AudioGainAudioProcessorEditor()
{
}


void AudioGainAudioProcessorEditor::paint(Graphics& g)
{
g.fillAll(getLookAndFeel().findColour(ResizableWindow::backgroundColourId));
}
```

```
void AudioGainAudioProcessorEditor::resized()
{
//Määritetään säätimen koko, tässä tapauksessa ikkunan koon mukaisesti
gainSlider.setBounds(getLocalBounds());
}

//Määritetään säätimen arvojen muuttumiseen tarvittavia toimintoja
void AudioGainAudioProcessorEditor::sliderValueChanged(Slider *slider)
{
		if (slider == &gainSlider)
		{
				processor.gainValue = gainSlider.getValue();
		}
}
```

Liite 2

PLUGINEDITOR.H

```
#pragma once
#include "../JuceLibraryCode/JuceHeader.h"
#include "PluginProcessor.h"

class AudioGainAudioProcessorEditor : public AudioProcessorEditor,
//Määritetään luokka säätimen kuuntelijalle
public Slider::Listener

{
public:
AudioGainAudioProcessorEditor(AudioGainAudioProcessor&);
~AudioGainAudioProcessorEditor();

void paint(Graphics&) override;
void resized() override;
//Määritetään säätimen arvojen muuttumiseen tarvittavia toimintoja
void sliderValueChanged(Slider* slider) override;

//Määritetään olio automaatiotiedon käyttämistä varten
ScopedPointer <AudioProcessorValueTreeState::SliderAttachment> sliderAttach;

private:
AudioGainAudioProcessor& processor;

//Määritetään säädin
Slider gainSlider;


JUCE_DECLARE_NON_COPYABLE_WITH_LEAK_DETECTOR
(AudioGainAudioProcessorEditor)
};
```

Liite 3

PLUGINPROCESSOR.CPP

```
#include "PluginProcessor.h"
#include "PluginEditor.h"

AudioGainAudioProcessor::AudioGainAudioProcessor()
#ifndef JucePlugin_PreferredChannelConfigurations
                : AudioProcessor(BusesProperties()
#if ! JucePlugin_IsMidiEffect
#if ! JucePlugin_IsSynth
                                .withInput("Input", AudioChannelSet::stereo(), true)
#endif
                                .withOutput("Output", AudioChannelSet::stereo(), true)
#endif
                ),
//Alustetaan olio
parameters(*this, nullptr)

#endif
{
//Määritetään arvoalue (minimi, maksimi)
NormalisableRange<float> gainRange(-48.0f, 0.0f);

//Määritetään parametrien konfigurointi
parameters.createAndAddParameter
(GAIN_ID, GAIN_NAME, GAIN_NAME, gainRange, 0.5f, nullptr, nullptr);

//Alustetaan olio
parameters.state = ValueTree("savedParams");
}

AudioGainAudioProcessor::~AudioGainAudioProcessor()
{
}

const String AudioGainAudioProcessor::getName() const
{
return JucePlugin_Name;
}

bool AudioGainAudioProcessor::acceptsMidi() const
{
#if JucePlugin_WantsMidiInput
return true;
#else
return false;
#endif
}
```

```
bool AudioGainAudioProcessor::producesMidi() const
{
#if JucePlugin_ProducesMidiOutput
return true;
#else
return false;
#endif
}
bool AudioGainAudioProcessor::isMidiEffect() const
{
#if JucePlugin_IsMidiEffect
return true;
#else
return false;
#endif
}

double AudioGainAudioProcessor::getTailLengthSeconds() const
{
return 0.0;
}

int AudioGainAudioProcessor::getNumPrograms()
{
return 1;
}

int AudioGainAudioProcessor::getCurrentProgram()
{
return 0;
}

void AudioGainAudioProcessor::setCurrentProgram(int index)
{
}

const String AudioGainAudioProcessor::getProgramName(int index)
{
return {};
}

void AudioGainAudioProcessor::changeProgramName(int index, const String&
newName)
{
}
```

```
void AudioGainAudioProcessor::prepareToPlay(double sampleRate, int samplesPerB-
lock)
{
//Määritetään kaava, joka määrittää desibeliasteikon ohjelmaan
previousGain = pow(10, *parameters.getRawParameterValue(GAIN_ID) / 20);
}

void AudioGainAudioProcessor::releaseResources()
{

}

#ifndef JucePlugin_PreferredChannelConfigurations

bool AudioGainAudioProcessor::isBusesLayoutSupported(const BusesLayout& layouts)
const
{
#if JucePlugin_IsMidiEffect
ignoreUnused(layouts);
return true;
#else
if (layouts.getMainOutputChannelSet() != AudioChannelSet::mono()
&& layouts.getMainOutputChannelSet() != AudioChannelSet::stereo())
return false;

#if ! JucePlugin_IsSynth
if (layouts.getMainOutputChannelSet() != layouts.getMainInputChannelSet())
return false;
#endif

return true;
#endif
}
#endif

void AudioGainAudioProcessor::processBlock(AudioSampleBuffer& buffer, MidiBuffer&
midiMessages)
{
ScopedNoDenormals noDenormals;
const int totalNumInputChannels = getTotalNumInputChannels();
const int totalNumOutputChannels = getTotalNumOutputChannels();
//Määritetään kaava, joka määrittää desibeliasteikon ohjelmaan
float currentGain = pow(10, *parameters.getRawParameterValue(GAIN_ID) / 20);

for (int i = totalNumInputChannels; i < totalNumOutputChannels; ++i)
buffer.clear(i, 0, buffer.getNumSamples());
```

```
//Määritetään lause, joka estää äänen häiriöt ohjelmassa
if (currentGain == previousGain)
{
buffer.applyGain(currentGain);
}

else
{
buffer.applyGainRamp(0, buffer.getNumSamples(), previousGain, currentGain);
previousGain = currentGain;
}
}

bool AudioGainAudioProcessor::hasEditor() const
{
return true;
}

AudioProcessorEditor* AudioGainAudioProcessor::createEditor()
{
return new AudioGainAudioProcessorEditor(*this);
}

void AudioGainAudioProcessor::getStateInformation(MemoryBlock& destData)
{
// Luetaan sen hetkinen tieto sekä luodaan XML-tiedosto
ScopedPointer <XmlElement> xml(parameters.state.createXml());
copyXmlToBinary(*xml, destData);
}

void AudioGainAudioProcessor::setStateInformation(const void* data, int sizeInBytes)
{
//Asetetaan ohjelman sen hetkinen tieto XML-tiedostoon
ScopedPointer <XmlElement> theParams(getXmlFromBinary(data, sizeInBytes));

//Määritetään lause virheentarkistusta varten
if (theParams != nullptr)
{
                if (theParams->hasTagName(parameters.state.getType()))
                {
                parameters.state = ValueTree::fromXml(*theParams);
                }
}
}
AudioProcessor* JUCE_CALLTYPE createPluginFilter()
{
                return new AudioGainAudioProcessor();
}
```

Liite 4

PLUGINPROCESSOR.H

```
#pragma once
#include "../JuceLibraryCode/JuceHeader.h"

//Määritetään makrot parametrien määrittämisen helpottamiseksi
#define GAIN_ID "gain"
#define GAIN_NAME "Gain"

class AudioGainAudioProcessor : public AudioProcessor
{
public:
AudioGainAudioProcessor();
~AudioGainAudioProcessor();
void prepareToPlay(double sampleRate, int samplesPerBlock) override;
void releaseResources() override;
#ifndef JucePlugin_PreferredChannelConfigurations
bool isBusesLayoutSupported(const BusesLayout& layouts) const override;
#endif
void processBlock(AudioSampleBuffer&, MidiBuffer&) override;
AudioProcessorEditor* createEditor() override;
bool hasEditor() const override;
const String getName() const override;
bool acceptsMidi() const override;
bool producesMidi() const override;
bool isMidiEffect() const override;
double getTailLengthSeconds() const override;
int getNumPrograms() override;
int getCurrentProgram() override;
void setCurrentProgram(int index) override;
const String getProgramName(int index) override;
void changeProgramName(int index, const String& newName) override;

void getStateInformation(MemoryBlock& destData) override;
void setStateInformation(const void* data, int sizeInBytes) override;

//Määritetään muuttuja
float gainValue;

//Määritetään olio
AudioProcessorValueTreeState parameters;

//Määritetään muuttuja
float previousGain;
private:
JUCE_DECLARE_NON_COPYABLE_WITH_LEAK_DETECTOR(AudioGainAudioProcessor)
};
```